

Query Management Facility™



QMF 응용프로그램 개발

버전 7

Query Management Facility™



QMF 응용프로그램 개발

버전 7

— 주! —

이 책과 이 책이 지원하는 제품을 사용하기 전에 307 페이지의 『부록G. 주의사항』에 나오는 일반 정보를 반드시 읽으십시오.

제7판(2000년 9월)

이 책은 후속판에 특별히 명시하기 전까지, Query Management Facility, OS/390용 DB2 Universal Database(OS/390용 DB2 UDB) 서버 버전 7 릴리스 1, 5675-DB2 및 Query Management Facility의 기능, VM 및 VSE용 DATABASE 2(VM 및 VSE용 DB2) 서버 버전 7 릴리스 1, 5697-F42의 기능과 모든 후속 릴리스 및 수정판에 적용됩니다. 제품 레벨에 따라 적절한 책을 사용하십시오.

이 책은 SA30-0660-00의 수정판입니다.

이 책의 기술적 변경사항은 변경사항의 왼쪽에 수직 막대로 표시되어 있습니다. 그림이 변경된 경우에는 그림 제목의 왼쪽에 수직 막대가 표시되어 있습니다. 기술적으로 중요하지 않은 편집상의 변경사항은 표시되어 있지 않습니다.

목차

QMF 라이브러리	vii	대체 변수가 없는 REXX 프로그램 시작	19
이 책에 대하여	ix	대체 변수가 있는 REXX 프로그램 호출	19
이 책의 사용 방법	ix	제3장 호출 인터페이스	21
시작하기 전에 알아야 할 사항	ix	호출 인터페이스란	21
고객 의견서 보내는 방법	x	QMF 호출 인터페이스를 사용하는 경우의	
QMF 서적 주문 방법	x	고려사항	22
제1장 QMF 응용프로그램 개발 개요	1	인터페이스 통신 영역의 정의(DSQCOMM)	23
QMF에서 응용프로그램 개발의 의미	1	리턴 코드	25
일반 사용자가 응용프로그램을 사용하는 방법	2	호출 인터페이스를 사용하는 경우의 명령	26
사용가능한 QMF 응용프로그램 개발 도구	4	응용프로그램에서 QMF 시작	27
QMF 프로시저	4	호출 인터페이스 응용프로그램 실행	27
QMF 호출 인터페이스 및 명령 인터페이스	5	QMF에서 호출 인터페이스 사용	28
QMF 오브젝트의 외부 형식	7	오류 처리	28
명령 동의어	7	CICS에서 호출 인터페이스 프로그램 실행	30
QMF에 연결되는 다른 IBM 제품	7	제4장 응용프로그램에 명령 인터페이스 사용	33
제2장 프로시저를 응용프로그램으로 사용	9	명령 인터페이스를 사용하는 프로그램 작성:	
프로시저를 사용하지 말아야 할 경우	9	예제	34
초기 프로시저	10	명령 인터페이스 호출	35
초기 프로시저 작성에 대한 고려사항	10	END 명령	36
초기 프로시저 및 원격 작업 단위(UOW)	11	명령 인터페이스에서 변수 사용	36
프로시저에서 QMF CONNECT 사용	11	명령 인터페이스 리턴 코드	38
프로시저의 대체 변수	12	0에서 16까지의 리턴 코드	39
RUN 명령에 값 지정	13	20 이상의 리턴 코드	40
RUN 명령 프롬프트 패널에 값 지정	13	제5장 ISPF를 사용하는 QMF 응용프로그램	
논리를 갖는 프로시저에서 REXX 변수 사용	15	작성	43
논리를 갖는 프로시저에 인수 전달	16	ISPF 응용프로그램에서 QMF의 시작 및 실행	43
논리를 갖는 프로시저에서 REXX 오류 처리		변수를 포함하는 조회 실행	44
명령문 사용	16	ISPF에서 QMF 논리를 갖는 프로시저로부터	
오류 처리 서브루틴으로 분기	16	프로그램 호출	45
REXX EXIT문에서 메시지 사용	17	논리를 갖는 프로시저로부터 ISPF 명령 사용	46
논리를 갖는 프로시저로부터 REXX 프로그램		호출 인터페이스 고려사항	47
호출	18	ISPF에서 EDIT 명령 사용	47





ISPF를 사용한 응용프로그램 디버그	48	REXX 호출 인터페이스의 시작 명령 구문	78
ISPF 로그 서비스 사용	48	START 명령 키워드	78
PDF 대화 테스트 사용	48	명령 동의어 사용	84
제6장 2개 언어 응용프로그램 작성	51	명령 동의어 작성	84
응용프로그램에 대한 2개 언어 오브젝트 작성	51	SAA RUN QUERY 보고서 소형 세션	85
명령 언어 변수 사용	52	제8장 QMF 오브젝트 가져오기 및 내보내기	89
2개 언어 응용프로그램에서 초기 프로시저 사		내보낸 파일, 데이터 세트 또는 CICS 데이터	
용	53	대기열로 할 수 있는 작업	90
영어 명령 사용	54	데이터 내보내기 및 저장	91
다국어 환경	54	데이터 및 표 오브젝트	92
QMF 세션 환경	55	QMF 형식의 데이터 오브젝트 해석: 예제	93
환경적 유사점	55	데이터 및 표 오브젝트의 내보내기/가져오기	
환경적 차이점	56	규칙과 정보	95
변역할 수 있는 응용프로그램 작성	57	프로시저 및 SQL 조회	97
제7장 응용프로그램의 QMF 명령	59	차트 오브젝트	98
CONNECT	59	코드화 오브젝트	98
예제	60	고정 형식 레코드	99
END	61	가변 형식 레코드	103
호출 인터페이스에 의해 세션 시작	62	응용프로그램 데이터 레코드(*)	111
초기 프로시저(DSQSRUN)가 있는 대화식		코드화 형식 오브젝트 내보내기	112
세션	62	코드화 형식 오브젝트 가져오기	112
초기 프로시저가 없는 대화식 세션	63	프롬프트 조회 오브젝트	113
INTERACT 명령에 의해 시작되는 대화식		프롬프트 조회 오브젝트 내보내기	114
세션	63	프롬프트 조회 가져오기	115
일괄처리 모드 세션	63	서식 오브젝트	117
EXIT	64	기본 서식 작성: 예제	117
GET GLOBAL	65	응용프로그램에서 QMF 서식 오브젝트에	
INTERACT	66	대한 고려사항	120
INTERACT의 세션 서식	66	서식 오브젝트 가져오기	123
INTERACT의 명령 서식	69	서식 오브젝트 내보내기	126
MESSAGE	70	보고서 오브젝트	127
MESSAGE 명령을 사용한 메시지 생성 예		보고서 내보내기 예제	128
제	72	보고서 행 레코드(L)	130
SET GLOBAL	73	데이터 연속 레코드(C)	132
SET GLOBAL: 선형 구문	74	HTML 보고서	135
SET GLOBAL: 확장 구문	74	QBE 조회	136
START	77	외부화 QMF 오브젝트의 세부사항	137

CICS 대기열 사용 규칙 및 고려사항	138	마이그레이션 정보	193
제9장 QMF 응용프로그램 디버깅	141	COBOL 프로그래밍 예제	194
호출 인터페이스 응용프로그램 디버깅	141	COBOL용 DSQCOMM	196
L 옵션을 사용한 추적	141	COBOL 호출 인터페이스 프로그램 실행 에 대한 고려사항	198
A 옵션을 사용한 추적	142	CICS에서 COBOL 프로그램 실행	199
추적 해제	143	VM의 CMS에서 프로그램 컴파일 및 실행	202
QMF 추적 데이터 출력 할당	144	TSO에서 COBOL 프로그램 실행	203
QMF 메시지 명령에 추적 사용	144	FORTRAN 언어 인터페이스	207
START 및 다른 QMF 명령의 오류 디버깅	146	FORTRAN의 인터페이스 통신 영역 맵핑 (DSQCOMM)	207
부록A. 호출 인터페이스 언어의 샘플 코드	147	FORTRAN의 함수 호출	208
어셈블러 언어 인터페이스	148	FORTRAN 프로그래밍 예제	210
어셈블러에 대한 인터페이스 통신 영역 맵 핑(DSQCOMM)	148	FORTRAN용 DSQCOMM	215
어셈블러 언어의 함수 호출	149	VM의 CMS에서 프로그램 컴파일 및 실행	217
마이그레이션 정보	151	MVS의 TSO에서 프로그램 실행	219
어셈블러 프로그래밍 예제	152	PL/I 언어 인터페이스	224
어셈블러용 DSQCOMM	162	PL/I의 인터페이스 통신 영역 맵핑 (DSQCOMML)	224
CICS에서 어셈블러 프로그램 실행	163	PL/I의 함수 호출	225
VM의 CMS에서 프로그램 어셈블 및 실행	166	MVS에서 CICS 사용자를 위한 마이그레이션 정보	227
TSO에서 어셈블러 프로그램 실행	168	PL/I 프로그래밍 예제	227
C 언어 인터페이스	171	PL/I용 DSQCOMM	231
C 언어의 인터페이스 통신 영역 맵핑 (DSQCOMM)	172	CICS에서 프로그램 실행	232
C 언어의 함수 호출	173	VM의 CMS에서 프로그램 컴파일 및 실행	235
마이그레이션 정보	175	TSO에서 컴파일 및 링크 편집	237
C 언어 프로그래밍 예제	176	REXX 언어 인터페이스	241
C용 DSQCOMM	180	REXX의 인터페이스 통신 변수	241
CICS에서 프로그램 실행	181	REXX에 대한 함수 호출	244
VM의 CMS에서 프로그램 컴파일 및 실행	184	REXX 프로그래밍 예제	245
TSO에서 C 프로그램 실행	186	VM의 CMS에서 프로그램 실행	247
COBOL 언어 인터페이스	189	MVS의 TSO에서 프로그램 실행	248
COBOL의 인터페이스 통신 영역 맵핑 (DSQCOMMB)	190	INTERACT 루프 사용에 대한 REXX 예 제	249
COBOL의 함수 호출	191		
COBOL에서 ISPF LIBDEF 서비스 사용	193		

부록B. 내보내기/가져오기 형식	251	명령 및 프로시저가 실행되는 방법을 제어하 는 DSQ 글로벌 변수	301
데이터에 대한 QMF 형식	251	CONVERT QUERY의 결과를 보여주는 DSQ 글로벌 변수	305
헤더 레코드.	251	RUN QUERY 오류 메시지 정보를 보여주는 DSQ 글로벌 변수	306
데이터 레코드.	253	부록G. 주의사항	307
프롬프트 조회 오브젝트의 표 및 필드 번호	254	등록상표.	310
서식 오브젝트의 표 및 필드 번호	256	용어집	311
보고서 오브젝트의 표와 필드 번호	261	참고 문헌	329
QMF 보고서에서 사용되는 HTML 태그	263	APPC 서적.	329
부록C. 통합 교환 형식(IXF)	265	CICS 서적	329
헤더 레코드(H)	266	COBOL 서적	330
표 레코드(T)	266	DATABASE 2 서적	330
열 레코드(C)	267	DCF 서적	331
데이터 레코드(D).	268	DRDA 서적	331
열 데이터 형식	268	DXT 서적	332
데이터 유형별 열 데이터 형식	269	그래픽 데이터 표시 관리자(GDDM) 서적	332
IXF의 예제.	276	HLASM 서적	332
부록D. ADDRESS QRW: QMF 명령 환 경 사용	281	ISPF/PDF 서적	332
부록E. 제품 인터페이스 매크로	283	OS/390 서적	333
부록F. QMF 글로벌 변수 표.	285	PL/I 서적	333
프로파일 관련 상태 정보에 대한 DSQ 글로 벌 변수	286	REXX 서적	334
프로파일과 관련없는 상태 정보에 대한 DSQ 글로벌 변수.	287	ServiceLink 서적.	334
CICS와 연관된 DSQ 글로벌 변수	292	VM 서적	334
이전 명령에 의해 생성된 메시지와 관련된 DSQ 글로벌 변수	294	VSE 서적	334
표 편집기와 연관된 DSQ 글로벌 변수.	294	색인	335
화면에 정보가 표시되는 방법을 제어하는 DSQ 글로벌 변수	297		

QMF 라이브러리

IBM 영업대표를 통해 메뉴얼을 주문할 수 있습니다.

평가	<p>QMF 소개서</p> <p>GA30-1047</p>			
설치, 계획 관리 및 진단	<p>Installing and Managing QMF on OS/390</p> <p>GC27-0719</p>	<p>Installing and Managing QMF on VM/ESA</p> <p>GC27-0720</p>	<p>Installing and Managing QMF on VSE/ESA</p> <p>GC27-0721</p>	<p>Installing and Managing QMF for Windows</p> <p>GC27-0722</p>
	<p>QMF 메시지 및 코드</p> <p>GA30-1050</p>	<p>QMF High Performance Option User's Guide for OS/390</p> <p>SC27-0724</p>		
사용	<p>QMF 사용법</p> <p>SA30-1049</p>	<p>QMF 참조서</p> <p>SA30-1048</p>	<p>Windows용 QMF 시작하기</p> <p>SA30-0663</p>	
응용 프로그램 프로그래밍	<p>QMF 개발 응용 프로그램</p> <p>SA30-1051</p>			
온라인 라이브러리	 <p>SK2T-0730 OS/390, VM, 및 VSE</p>	 <p>SK2T-6700 OS/390 전용</p>	 <p>SK2T-2067 VM 전용</p>	 <p>SK2T-0060 VSE 전용</p>

이 책에 대하여

이 책은 IBM® QMF(Query Management Facility)를 사용하는 응용프로그램을 작성하는 응용프로그램 프로그래머를 돕기 위해 쓰여진 것입니다.

이 책의 사용 방법

이 책에 포함된 타스크들은 QMF 응용프로그램을 작성하기 전에 수행해야 할 설계 결정사항을 대략 설명하고, 다양한 프로그래밍 기술을 보여주고, QMF를 사용하는 응용프로그램 프로그래밍에 초점을 둔 몇 개의 예제를 제공합니다. 부록에서는 응용프로그램 개발에 유용한 참조 정보를 제공합니다.

이 책은 OS/390®, VM 및 VSE 사용자에게 제공됩니다. 필요에 따라 각 시스템 간 차이점이나 CICS, CMS, TSO 및 고유 OS/390 일괄처리간 차이점이 강조표시됩니다. 그렇지 않으면, QMF가 각 시스템에서 동일한 방식으로 작동한다고 가정하면 됩니다.

시작하기 전에 알아야 할 사항

QMF 응용프로그램은 QMF 오브젝트로 작업할 수 있게 하고, QMF가 지원하는 언어 중 하나로 작성된 응용프로그램에서 QMF 기능을 수행하게 합니다. 이 책에서는 사용자가 조회 및 프로시저 작성법, 보고서의 형식 지정 방법, 데이터베이스 수정 방법을 이미 알고 있다고 가정합니다.

QMF 명령 또는 호출 인터페이스를 사용하여 QMF 응용프로그램을 작성하려면, 다음 프로그래밍 언어 중 하나를 알고 있어야 합니다.

호출 인터페이스

명령 인터페이스

어셈블러, PL/I, C, REXX, COBOL, FORTRAN

ISPF에서 실행되는 모든 언어

이 책에 대하여

작성할 응용프로그램 유형에 따라 패널 표시 응용프로그램이 필요할 수도 있습니다.

QMF 기능 및 관리에 관한 정보를 제공하는 서적 목록을 보려면 vii 페이지의 『QMF 라이브러리』를 참조하십시오.

고객 의견서 보내는 방법

고객의 의견은 가장 정확하고 고품질의 정보를 제공하는 데 있어 중요한 역할을 합니다.

웹을 통해 의견 보내기

다음 주소의 웹 사이트를 방문하십시오.

<http://www.ibm.com./qmf>

이 웹 사이트에는 고객이 의견을 입력하고 보낼 수 있는 피드백 페이지가 있습니다.

고객 의견서 서식 작성

이 책 맨 뒤에 있는 서식을 작성하여 팩스로 보내 주십시오.

QMF 서적 주문 방법

IBM 영업대표를 통해 QMF 서적을 주문할 수 있습니다.

QMF 서적 목록은 vii 페이지의 『QMF 라이브러리』를 참조하십시오.

제1장 QMF 응용프로그램 개발 개요

자체 응용프로그램에서 많은 QMF 기능을 사용할 수 있습니다. 예를 들어 다음 기능을 가지는 응용프로그램을 작성할 수 있습니다.

- 조회 또는 프로시저를 실행합니다.
- QMF 오브젝트를 내보내거나 가져옵니다.
- 보고서 또는 차트를 표시하거나 인쇄합니다.
- 사용자가 데이터베이스에 있는 데이터를 입력하거나 변경할 수 있게 합니다.

또한 원격 위치에 있는 QMF 보고서를 인쇄하는 사용자 정의 명령이나 주간 판매 차트를 자동으로 생성하는 기능 키처럼 사용자에게 도움을 줄 수 있는 기능을 제공하는 응용프로그램도 QMF를 통해 작성할 수 있습니다.

이 장에서는 QMF 응용프로그램의 두 가지 기본 유형과 응용프로그램을 구현하는데 도움을 주기 위해 QMF가 제공하는 응용프로그램 개발 도구에 대해 설명합니다.

QMF에서 응용프로그램 개발의 의미

응용프로그램이라는 단어에는 많은 의미가 있습니다. QMF에서 응용프로그램은 프로시저, 프로그램 또는 QMF 명령을 실행하는 EXEC일 수 있으며, 내보내기 및 가져오기 QMF 명령을 사용하여 QMF 오브젝트를 변경하기도 합니다.

응용프로그램 개발이란 응용프로그램을 작성하는 프로세스를 뜻합니다. 이 프로세스에는 다음과 같은 작업이 포함됩니다.

- 응용프로그램이 해결하는 문제점 이해
- 응용프로그램 설계
- 코드, 관련 메시지 및 도움말 패널 작성

이러한 작업이 완료되었다면, 일반 사용자가 응용프로그램을 사용하는 방법과 응용프로그램이 QMF와 대화할 수 있게 하기 위해 사용해야 할 QMF 응용프로그램 개발 도구를 결정할 수 있게 됩니다.

일반 사용자가 응용프로그램을 사용하는 방법

일반 사용자가 1차적으로 응용프로그램과 대화하도록 할 수도 있고 QMF에서 사용자 정의된 기능으로서 응용프로그램을 사용하도록 할 수도 있습니다.

- 응용프로그램이 QMF에 익숙하지 않은 일반 사용자를 대상으로 하고 있다면 일반 사용자가 1차적으로 응용프로그램과 대화하기를 원할 것입니다. 사실상, 일반 사용자가 QMF가 사용중이라는 것을 모르게 할 수도 있습니다. 이러한 경우, 응용프로그램은 QMF 서비스를 사용하지만 QMF 외부에 상주합니다. 프로그램은 필요한 경우에만 QMF 명령을 발행합니다.
- 일반 사용자가 QMF에 익숙하다면 응용프로그램을 QMF의 확장 기능 또는 조정 기능으로 다루게 할 수도 있습니다. 이러한 경우, QMF에서 실행되도록 응용프로그램을 설정해야 합니다.

일반 사용자가 1차적으로 응용프로그램과 대화할 경우

사용자가 QMF 서비스를 사용하는 응용프로그램을 작성한다고 가정해 보십시오. 그림1과 같이 이 응용프로그램은 일반 사용자에게 메뉴 방식의 인터페이스를 제공합니다.

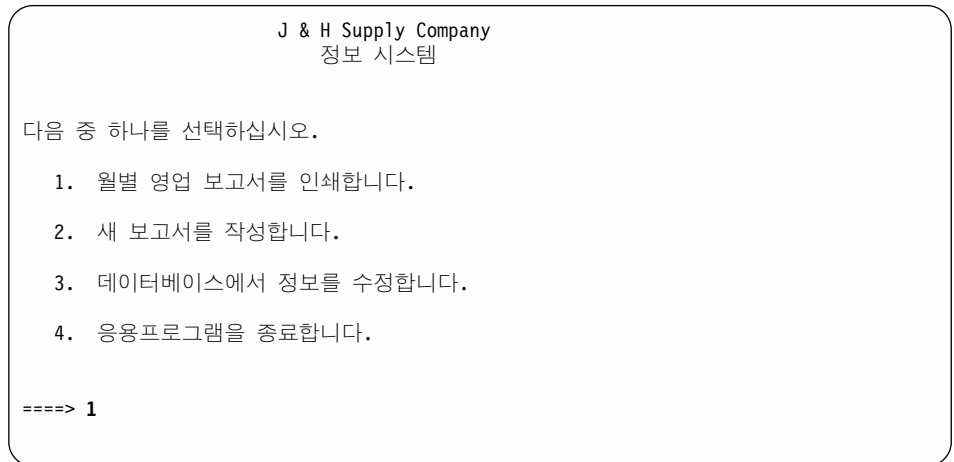


그림 1. 응용프로그램으로 정의된 패널 예제

사용자가 옵션을 선택하면 응용프로그램은 적절한 QMF 명령을 발행합니다. 예를 들어 사용자가 옵션 1을 선택하면 응용프로그램은 조회를 실행하고 결과 보고서를 인쇄할 수도 있는 QMF 프로시저를 실행합니다.

앞의 예제에서는 응용프로그램은 QMF를 제어합니다. 사용자는 사용자 인터페이스와만 상호 작용하며 QMF가 사용중인지는 인식하지 못합니다.

일반 사용자가 QMF 세션에서 응용프로그램을 시작하는 경우

QMF 보고서를 한 사용자에서 다른 사용자에게 보내는 응용프로그램을 작성한다고 가정해 보십시오.

사용자가 QMF 환경에서 응용프로그램을 실행하기를 기대하므로 일반 사용자가 명령행에서 발행할 수 있는 명령 동언어를 응용프로그램에 할당하거나 자동으로 사용자의 응용프로그램을 실행하는 응용프로그램을 기능 키에 할당할 수 있습니다.

사용자는 보고서를 생성한 후 4 페이지의 그림2와 같이 QMF 명령행에 SEND_TO SMITH를 입력하여 이 보고서를 Smith에게 보낼 수 있습니다.

```

보고서                                행 1      POS 1      79

NAME      DEPT  JOB      SALARY      COMM
-----
DANIELS   10  MGR      19260.25    -
JONES     10  MGR      21234.00    -
LU        10  MGR      20010.00    -
MOLINARE  10  MGR      22959.20    -
HANES     15  MGR      20659.80    -
KERMISCH  15  CLERK    12258.50    110.10
NGAN      15  CLERK    12508.20    206.60
ROTHMAN   15  SALES    16502.83    1152.00
JAMES     20  CLERK    13504.60    128.20
PERNAL    20  SALES    18171.25    612.45
SANDERS   20  MGR      18357.50    -
SNEIDER   20  CLERK    14252.75    126.50
ABRAHAMS  38  CLERK    12009.75    236.50
MARENGHI  38  MGR      17506.75    -
1=도움말      2=      3=종료      4=인쇄      5=차트      6= 조회
7=앞화면      8=뒤화면  9=서식      10=왼쪽     11=오른쪽    12=
여기에 보고서가 있습니다.
명령 ==> SEND_TO SMITH
    
```

그림 2. 사용자 정의된 QMF 명령을 입력하는 사용자 예제

사용가능한 QMF 응용프로그램 개발 도구

일반 사용자가 응용프로그램을 보는 방법에 관계없이 다음 응용프로그램 개발 도구를 사용하여 응용프로그램을 작성할 수 있습니다.

- QMF 프로시저
- QMF 호출 인터페이스
- QMF 명령 인터페이스
- QMF 외부
- QMF 명령 동의어
- QMF에 연결되는 다른 IBM 제품

QMF 프로시저

QMF 프로시저는 QMF에서 실행되고 QMF 명령을 발행하는 QMF 오브젝트입니다. QMF 프로시저는 설치에 사용할 수 있는 모든 QMF 명령을 실행할 수 있습니다. QMF는 선형 프로시저와 논리를 갖는 프로시저의 두 가지 유형의 프로시저를 제공합니다.

선형 프로시저는 QMF 명령과 주석만을 포함합니다. QMF에서 지원되는 모든 환경에서 선형 프로시저를 사용할 수 있습니다.

논리를 갖는 프로시저는 좀더 강력한 프로그램을 작성할 수 있게 하기 위해 QMF 명령을 REXX 논리와 결합합니다. CICS®를 제외하고 QMF에서 지원되는 모든 환경에서 논리를 갖는 프로시저를 사용할 수 있습니다. 논리를 갖는 프로시저는 QMF 명령과 REXX 프로그램에서 유효한 모든 명령문을 포함할 수 있습니다.

선형 프로시저 또는 논리를 갖는 프로시저에 대한 일반적인 정보는 *QMF 사용법*을 참조하십시오. QMF 프로시저를 사용하여 응용프로그램을 작성하는 방법에 대한 특정 정보는 9 페이지의 『제2장 프로시저를 응용프로그램으로 사용』을 참조하십시오.

버전 3.3에서 작업을 시작하면 QMF는 사용자가 QMF 시작 패널을 보기 전에 명령을 실행하고 글로벌 변수를 설정할 수 있게 해주는 시스템 초기화 프로시저를 제공합니다. 자세히 알려면 *Installing and Managing QMF* 버전을 참조하십시오.

QMF 호출 인터페이스 및 명령 인터페이스

QMF 프로시저를 사용하지 않기로 한 경우, 프로그램이 호출 인터페이스와 명령 인터페이스 중 어느 인터페이스를 통해 QMF와 통신할지 결정해야 합니다.

호출 인터페이스

QMF 호출 인터페이스는 시스템 응용프로그램 구조(SAA) 인터페이스로서 QMF 외부에서 호출되고, QMF 세션을 시작하고, QMF로 명령을 보내 실행되게 하는 응용프로그램을 작성하는 데 사용됩니다.

호출 인터페이스는 QMF에서 지원되는 모든 환경에서 사용할 수 있습니다. 이것은 VM, OS/390 및 VSE 환경에서 조희를 위한 SAA 공통 프로그래밍 인터페이스이며, 표1에 표시된 대로 다양한 언어에 사용할 수 있습니다.

표 1. 호출 인터페이스 지원

	OS/390의 CICS	VSE의 CICS	CMS	TSO	APPC	SRPI	고유 OS/390 일괄처리
어셈블러 ¹	x	x	x	x	x	x	x

QMF 응용프로그램 개발 개요

표 1. 호출 인터페이스 지원 (계속)

	OS/390의 CICS	VSE의 CICS	CMS	TSO	APPC	SRPI	고유 OS/390 일괄처리
C	x	x	x	x	x	x	x
COBOL	x	x	x	x	x	x	x
FORTRAN			x	x	x	x	x
PL/I	x	x	x	x	x	x	x
REXX			x	x	x	x	x

SAA 응용프로그램을 작성하려면 QMF가 지원하는 SAA 언어 중 하나에서 호출 인터페이스를 사용해야 합니다.

호출 인터페이스에 관한 자세한 내용은 21 페이지의 『제3장 호출 인터페이스』를 참조하십시오.

명령 인터페이스

QMF 명령 인터페이스는 ISPF 대화로부터 QMF에 명령을 제공하는 응용프로그램을 작성할 수 있게 합니다. QMF는 이 명령 인터페이스를 사용하여 ISPF 변수 풀을 통해 ISPF 대화와 통신합니다.

명령 인터페이스는 ISPF를 사용할 수 있는 경우에만 사용할 수 있습니다. CICS에서는 명령 인터페이스를 사용할 수 없습니다.

QMF 명령 인터페이스에 관한 자세한 내용은 33 페이지의 『제4장 응용프로그램에 명령 인터페이스 사용』을 참조하십시오.

호출 인터페이스와 명령 인터페이스의 비교

호출 인터페이스와 명령 인터페이스간 차이는 다음과 같습니다.

호출 인터페이스

- QMF가 지원하는 모든 환경에서 사용할 수 있습니다.
- ISPF를 요구하지 않습니다.
- 응용프로그램을 실행하기 전에 QMF를 시작할 필요가 없습니다.
- 조회를 위한 SAA 공통 프로그래밍 인터페이스를 제공합니다.

1. 어셈블러는 SAA 언어가 아닙니다.

명령 인터페이스

- QMF 및 ISPF에서 지원되는 모든 환경에서 사용할 수 있습니다.
- ISPF가 존재해야 하며 활성 상태여야 합니다.
- 응용프로그램이 시작되기 전에 QMF가 시작되어야 합니다.
- ISPF 응용프로그램과 QMF 사이의 통신용 변수를 제공합니다.
- ISPF가 프로그래밍 언어를 지원해야 합니다.

QMF 오브젝트의 외부 형식

응용프로그램은 QMF 오브젝트를 QMF 제품의 파일 외부로 내보낼 수 있습니다. 예를 들어 서식을 CMS 파일, TSO 데이터 세트 또는 CICS 데이터 대기열로 내보낼 수 있습니다. 각 오브젝트에는 응용프로그램이 편집하고 다른 시스템으로 전송하거나 QMF로 가져올 때 사용하는 특별한 형식을 가집니다.

QMF 오브젝트의 구체적인 형식에 관한 자세한 내용은 89 페이지의 『제8장 QMF 오브젝트 가져오기 및 내보내기』를 참조하십시오.

명령 동의어

QMF는 사용자가 코드화하는 프로그램이나 프로시저에 대한 명령 동의어를 지정할 수 있게 해줍니다. 이 명령 동의어는 일반 사용자가 QMF 명령을 사용하는 것처럼 프로그램이나 프로시저를 사용하게 해줍니다.

명령 동의어에 관한 자세한 내용은 84 페이지의 『명령 동의어 사용』을 참조하십시오.

QMF에 연결되는 다른 IBM 제품

QMF와 다음 IBM 제품을 함께 사용하여 QMF의 기능을 확장할 수 있습니다.

응용프로그램 시스템(AS)

AS는 QMF 명령을 발행할 수 있고 QMF 조회를 정의할 수 있습니다. 그러면 AS는 QMF 조회의 결과를 AS 프로세스로 입력된 내용처럼 사용할 수 있습니다.

QMF 응용프로그램 개발 개요

데이터 추출(DXT)

QMF는 일반 사용자가 QMF에서 직접 지원하지 않는 원본으로부터 데이터를 추출할 수 있게 하기 위해 DXT™ 일반 사용자 대화를 호출할 수 있습니다.

ECF 향상된 연결성 기능(ECF)은 워크스테이션 사용자가 호스트 관계형 데이터를 액세스할 수 있도록 해줍니다. 워크스테이션 사용자는 ECF 기능을 사용하여 저장된 QMF 조회 또는 프로시저를 실행하고 검색된 데이터를 워크스테이션에 다운로드하도록 호스트에 요청을 보냅니다.

GDDM

차트를 표시하기 위해 QMF가 사용하는 대화식 차트 유틸리티(ICU)는 실제로 그래픽 데이터 표시 관리자(GDDM®)의 기능입니다.

ISPF 대화식 시스템 기능은 사용자가 명령 인터페이스를 통해 QMF와 대화할 수 있는 패널을 생성할 수 있게 해 줍니다.

Lotus 1-2-3/M

Lotus 1-2-3®의 호스트 버전은 QMF에 액세스하여 조회 결과에 대한 스프레드시트 분석을 수행할 수 있습니다.

제2장 프로시저를 응용프로그램으로 사용

많은 응용프로그램을 완전히 프로시저로 작성할 수 있습니다. 개발 시스템에서 프로시저를 작성할 수 있으며 개인적 용도로 보관하거나, 공공 용도로 생산 시스템으로 가져갈 수도 있습니다.

CICS 환경에서 QMF를 사용할 경우 QMF 선형 프로시저를 사용할 수 있습니다. CMS, TSO 또는 고유 OS/390 일괄처리 환경에서 QMF를 사용할 경우 QMF 프로시저에서 REXX문과 함수도 역시 사용할 수 있습니다. REXX 함수와 논리를 갖는 프로시저는 QMF CICS 환경에서는 사용할 수 없습니다.

이 장에서는 QMF 프로시저를 사용하여 응용프로그램을 구현하기 위해 알아야 할 정보에 초점을 맞추었습니다.

프로시저를 작성하고, 빌드하고, 실행하는 방법에 관한 내용은 *QMF 사용법*을 참조하십시오.

QMF 프로시저에서 ISPF 서비스를 사용하려면 몇 개의 단계가 더 필요합니다. ISPF에서 실행되는 QMF 논리를 갖는 프로시저로부터 ISPF 명령을 실행하는 데 대한 내용은 46 페이지의 『논리를 갖는 프로시저로부터 ISPF 명령 사용』을 참조하십시오.

프로시저를 사용하지 말아야 할 경우

프로시저에서 작동하는 응용프로그램을 QMF 임시 기억영역에 작성하는 경우 응용프로그램을 프로시저로서 작성할 수 없습니다. 한 프로시저를 실행하면 해당 프로시저가 QMF 임시 기억영역에서 현재 프로시저가 되기 때문입니다.

예를 들어, 프로시저로서 응용프로그램을 작성하고 QMF 임시 기억영역에 현재 프로시저를 저장하도록 응용프로그램을 코드화하면 응용프로그램이 실행될 경우 QMF 임시 기억영역에서 이 응용프로그램이 현재 프로시저가 되므로 응용프로그램은 자신을 저장합니다.

초기 프로시저

초기 프로시저는 QMF 세션이 시작된 후에 즉시 실행되는 프로시저입니다. 이 프로시저의 이름을 지정하려면 DSQSRUN 매개변수를 사용하십시오. DSQSRUN을 사용할 수 있습니다.

- QMF는 대화식으로 시작될 때 DSQQMFE 명령과 함께
- QMF가 호출 인터페이스를 통해 시작될 때 QMF START 명령과 함께

QMF는 사용된 QMF 세션의 유형에 따라 다르게 초기 프로시저를 실행합니다. QMF가 초기 프로시저를 사용하는 방법에 관한 자세한 내용은 62 페이지의 『초기 프로시저(DSQSRUN)가 있는 대화식 세션』을 참조하십시오.

TSO 및 고유 OS/390 일괄처리에서 응용프로그램은 REXX EXEC를 사용하는 프로그램 매개변수도 설정할 수 있습니다(QMF START 명령의 DSQSCMD 매개변수에 의해 설명됨). QMF CICS가 REXX를 지원하지 않기 때문에 CICS에서는 DSQSMODE=INTERACTIVE를 사용하여 START 명령에 모든 프로그램 매개변수를 지정해야 합니다. 호출 인터페이스의 기본 모드는 BATCH입니다.

초기 프로시저 작성에 대한 고려사항

- 기본적으로, DSQQMFE에 의해 시작되는 QMF의 대화식 세션에서 사용자가 END 명령을 발행할 때마다 QMF는 초기 프로시저를 다시 실행합니다. DSQEC_RERUN_IPROC 글로벌 변수는 초기 프로시저의 재실행 여부를 지정합니다. 이 변수의 기본값은 프로시저를 다시 실행하는 1이고, 0은 초기 프로시저가 다시 실행되지 못하게 합니다.

호출 인터페이스 프로그램에서 초기 프로시저는 결코 다시 실행되지 않으며 이 글로벌 변수도 호출 인터페이스 프로그램에 영향을 주지 않습니다.

- 대화식 QMF 세션에서 사용할 초기 프로시저를 작성하려면 프로시저의 끝에 있는 현재 패널이 시작 패널이 되도록 초기 프로시저를 작성하지 않아야 합니다. 시작 패널이 초기 프로시저의 끝에서 현재 패널이면 QMF는 프로시저의 끝에서 패널을 대화식으로 표시하지 않습니다. 심각한 오류가 발생하지 않고 DSQEC_RERUN_IPROC가 1로 설정되면, QMF에서는 사용자와의 대화 없이 초기 프로시저를 다시 실행합니다. 그 결과 QMF가 시작되지 않는 것처럼 보일 수 있는 연속 루프가 발생합니다.

연속 루프를 작성하지 않으려면, 다음 중 하나를 선택하십시오.

- 초기 프로시저의 끝에서 현재 패널이 시작 패널이 아닌지 확인합니다.
- 프로시저에 QMF EXIT 또는 INTERACT 명령 중 하나가 있는지 확인합니다.
- DSQEC_RERUN_IPROC를 0으로 설정합니다.
- 초기 프로시저에서 대체 변수에 대한 값을 지정할 경우 변수의 이름 앞에 사용해야 하는 앰퍼샌드(&)의 수는 환경에 따라 달라집니다. 예를 들면, 다음과 같이 DSQSRUN을 지정할 수 있습니다.

```
DSQSRUN=INITPROC(&VAR1 = value)
```

VAR1로 지정해야 하는 앰퍼샌드의 수는 QMF가 CICS, CMS, TSO 또는 고유 OS/390 일괄처리에서 실행되는지, ISPF가 있는지, QMF를 시작하는 프로그램이 REXX로 작성되었는지 여부에 따라 달라집니다. CMS 환경에 필요한 앰퍼샌드 수에 대한 자세한 내용은 *Installing and Managing QMF for VM/ESA* 를 참조하십시오. CICS/OS/390 또는 TSO에서는 *Installing and Managing QMF for MVS*를 참조하십시오. CICS/VSE®에서의 변수 값 전달에 대해서는 *Installing and Managing QMF for VSE/ESA* 내용을 참조하십시오.

초기 프로시저 및 원격 작업 단위(UOW)

초기 프로시저는 QMF를 시작하는 시스템(지역 시스템)에 저장되어야 합니다.

초기 프로시저로 설정된 대화식 세션중에 초기 프로시저 또는 명령행에서 QMF CONNECT 명령을 사용하면 END 명령을 사용하여 초기 프로시저를 재호출하기 전에 원래 위치에 다시 연결해야 합니다.

여전히 원격 위치로 연결되어 있다면 오류가 발생합니다.

프로시저에서 QMF CONNECT 사용

QMF CONNECT 명령으로 또다른 사용자 ID 또는 원격 DB2® 데이터베이스에 연결하여 원격 작업 단위 지원을 사용할 수 있습니다. 선형 프로시저 내에서 또는 논리를 갖는 프로시저와 함께 이 명령을 사용할 수 있습니다.

QMF CONNECT 명령을 사용하는 프로시저를 작성하여 원격 데이터베이스에 액세스할 경우, 다음 사항에 유의하십시오.

- 원격 데이터베이스와 연결된 상태에서 RUN PROC 명령을 발행하면 해당 프로시저에서 사용되는 프로시저와 모든 오브젝트는 원격 데이터베이스에 저장되어 있어야 합니다.
- 프로시저의 모든 QMF 명령은 QMF가 실행되고 있는 시스템의 QMF 임시 기억영역(지역 시스템)에서 실행됩니다. 그러나 QMF 명령에 의해 사용되는 모든 오브젝트(예: 조회, 프로시저, 서식)는 현재 위치(원격 시스템)의 데이터베이스에서 정의되어야 합니다.

QMF CONNECT 명령과 원격 작업 단위 지원에 대한 자세한 내용은 *QMF 참조서*(명령 구문의 경우)를 참조하십시오.

- 데이터베이스에 영향을 주는 모든 명령(예: SQL문, QMF 조회 또는 EDIT TABLE 갱신)은 현재 위치에서 실행됩니다.
- 프로시저에 시스템 고유 명령(CICS, CMS 또는 TSO)이 있으면 이 명령은 QMF가 실행되고 있는 시스템(지역 시스템)에서 실행됩니다.

프로시저에 QMF가 실행되는 시스템에서 실행되지 않는 시스템 고유 명령이 있으면 프로시저는 성공적으로 실행되지 않습니다.

- 시스템 고유 명령에 사용되는 모든 파일이나 데이터 세트가 QMF가 실행되고 있는 시스템(지역 시스템)에 존재해야 합니다.

프로시저의 대체 변수

선형 프로시저와 논리를 갖는 프로시저에서 QMF 대체 변수를 사용할 수 있습니다.

대체 변수는 QMF 명령에 사용할 수 있는 변수입니다. QMF가 이 변수를 관리합니다. 대체 변수 앞에는 항상 앰퍼샌드(&)가 옵니다. 글로벌 변수를 설정하거나, RUN 명령에 값을 지정하거나, RUN 명령 프롬프트 패널에 값을 지정하여 값을 대체 변수로 할당할 수 있습니다. 글로벌 변수 설정에 관한 내용은 73 페이지의 『SET GLOBAL』을 참조하십시오.

QMF에서 앰퍼샌드와 대체 변수 사용에 대해 자세히 알려면 *QMF 사용법*을 참조하십시오.

RUN 명령에 값 지정

RUN 명령을 사용하여 대체 변수에 값을 지정할 수 있습니다.

선형 프로시저의 경우:

```
RUN PROC SCHEDULE (&&TYPE='VACATION'
```

논리를 갖는 프로시저의 경우:

```
"RUN PROC SCHEDULE (&&TYPE='VACATION'"
```

PROC 또는 QUERY 패널 내에서 QMF RUN 명령을 발행하면, PROC 또는 QUERY 오브젝트 유형을 지정하지 않아도 됩니다. RUN은 각각의 패널에서 호출할 때 이러한 값을 가정합니다.

&TYPE의 값은 SCHEDULE로 불리는 프로시저에서만 사용할 수 있습니다.

이 예제에서는 다음이 적용됩니다.

- 변수 값 VACATION은 값이 하나의 문자열이기 때문에 작은 따옴표로 묶여 있습니다.
- SCHEDULE이라는 이름의 프로시저에 전달될 RUN문에 값이 설정되고 있음을 나타내기 위해 TYPE 앞에 이중 앰퍼샌드(&&) 기호가 붙습니다. RUN문이 &TYPE를 지정하면 이 명령문을 포함하는 프로시저는 사용자에게 값을 입력하라는 메시지를 표시합니다.

대체 변수에 대한 이 값은 그값을 정의한 프로시저에서만 사용됩니다. 값은 정의 중인 프로시저에서 호출된 어떠한 프로시저나 모듈에서도 사용되지 않습니다.

RUN 명령 프롬프트 패널에 값 지정

대체 변수를 포함하는 조회 또는 프로시저를 실행하고 있으며 글로벌 변수나 RUN 명령에 의해 이 변수에 값이 할당되지 않은 경우 QMF는 RUN 명령 프롬프트 패널을 표시합니다. 이 패널에서 변수 값을 지정할 수 있습니다.

대체 변수에 대한 이 값은 값을 정의한 프로시저에서만 사용됩니다. 값은 정의 중인 프로시저에서 호출된 어떠한 프로시저나 모듈에서도 사용되지 않습니다.

선형 프로시저의 변수 프롬프트

선형 프로시저에서 QMF는 프로시저에서 대체 변수를 스캔하고 분석한 뒤에 명령을 처리합니다. 모든 변수에 대해 사용자에게 프롬프트가 제시된 후에 프로시저가 실행됩니다.

논리를 갖는 프로시저의 변수 프롬프트

논리를 갖는 프로시저에서는 REXX가 변수를 포함하는 명령문이 나타날 때까지 사용자에게 변수에 대한 프롬프트가 제시되지 않습니다. 예를 들어, 논리를 갖는 프로시저에 QMF에서 프롬프트를 표시해야 하는 변수가 들어 있는 세 개의 명령문이 포함될 경우, QMF에서는 각 명령문마다 한 번씩 세 번 프롬프트를 표시합니다.

논리를 갖는 프로시저가 필요한 모든 변수 값에 대해 한번에 프롬프트를 표시하게 하려면 선형 프로시저의 가상 프로시저를 사용하십시오. 그림3에서처럼 논리를 갖는 프로시저에서 두 개의 행에 나타나는 대체 변수 LASTNAME 및 DEPT_NUM에 대해 한번만 프롬프트를 표시하려 한다고 가정해 보십시오.

```
/* This procedure runs two queries, displaying the report after each */  
/* procedure has run. */  
  
"RUN QUERY REG_QUERY (&&LASTNAME=&LASTNAME";  
"INTERACT"  
"RUN QUERY REG2_QUERY (&&DEPT_NUM=&DEPT_NUM";
```

그림 3. 변수가 있는 논리를 갖는 프로시저

다음 행을 주석행 바로 다음에 오는 논리를 갖는 프로시저의 시작 부분에 추가하십시오.

```
"RUN PROC PROMPT_ME (&LASTNAME, &DEPT_NUM";
```

여기서, PROMPT_ME는 15 페이지의 그림4에 표시된 대로, 주석행만 있고 명령어는 없는 논리를 갖는 프로시저입니다.

완료된 논리를 갖는 프로시저는 다음과 같습니다.

```

/* This proc is a dummy proc that provides prompting. */
/* This procedure runs two queries, displaying the report after each */
/* procedure has run */

"RUN PROC PROMPT_ME (&LASTNAME, &DEPT_NUM";
"RUN QUERY REG_QUERY (&&LASTNAME=&LASTNAME";
"INTERACT"
"RUN QUERY REG2_QUERY (&&DEPT_NUM=&DEPT_NUM";

```

그림 4. 변수에 대한 프롬프트를 표시하는 논리를 갖는 프로시저

또는 SET GLOBAL을 사용하여 다음과 같이 동시에 프로시저의 모든 값에 대한 프롬프트를 표시할 수 있습니다

```
"SET GLOBAL (LASTNAME=&LASTNAME,DEPTNUM=&DEPT_NUM";
```

논리를 갖는 프로시저에서 REXX 변수 사용

논리를 갖는 프로시저에서 REXX 변수를 사용할 수 있습니다. 이 변수에 대한 값은 해당 값이 정의된 프로시저에서만 알려져 있습니다. 다음을 수행할 수 있습니다.

- SET GLOBAL 명령으로 REXX 변수를 QMF 변수에 복사
- GET GLOBAL 명령으로 글로벌 변수를 REXX 변수에 복사
- REXX문에서 REXX 변수 사용

REXX 변수에 관한 자세한 내용은 시스템에 대한 REXX 참조 매뉴얼을 참조하십시오. GET GLOBAL 및 SET GLOBAL 명령에 관한 자세한 내용은 QMF 참조서를 참조하십시오.

QMF는 각 QMF 명령이 처리된 후 REXX 변수의 그룹을 QMF가 설정한 SAA 호출 인터페이스에 제공합니다. 이 변수는 각 명령의 결과에 관한 중요한 정보를 제공합니다. 논리를 갖는 프로시저에서 이러한 변수를 사용할 수 있습니다. 이 변수에 관한 자세한 내용은 241 페이지의 『REXX 언어 인터페이스』를 참조하십시오.

논리를 갖는 프로시저에 인수 전달

논리를 갖는 프로시저의 경우 QMF는 RUN PROC 명령에 ARG 옵션을 제공합니다. 이 옵션은 인수 또는 값을 논리를 갖는 프로시저로 전달하게 합니다.

다음 예제에서와 같이 REXX PARSE ARG 또는 ARG 문이 들어 있는 프로시저를 실행할 경우, ARG 옵션을 사용하십시오.

```

PROC                WILDE.SHOW_ARGS                MODIFIED    LINE 1

/*****
/* This procedure shows you how to use the 'ARG=' option on the RUN    */
/* PROC command.                                                    */
/*****
parse upper arg query_name form_name
"RUN QUERY" query_name "(FORM="form_name

```

이 프로시저에 대한 RUN 명령은 다음과 같습니다.

```
RUN PROC SHOW_ARGS (ARG=(query_name form_name))
```

ARG 옵션을 사용하면 프로시저간에도 값을 전달할 수 있습니다.

논리를 갖는 프로시저에서 REXX 오류 처리 명령문 사용

논리를 갖는 프로시저에서 REXX SIGNAL 명령과 같은 REXX 오류 처리 기술을 사용할 수 있습니다. 그리고 QMF 명령 및 변수를 REXX EXIT 명령과 함께 사용하면 0이 아닌 리턴 코드를 명확히 표현하는 데 도움이 될 수 있습니다.

오류 처리 서브루틴으로 분기

오류에 대한 REXX 신호는 0이 아닌 리턴 코드가 발생한 경우 REXX에 현재 행을 구분하고 오류가 표시된 레이블로 분기하라고 지시합니다. 이 명령문에는 다음과 같은 두 부분이 필요합니다.

- 오류에 대한 신호

REXX는 모든 명령 뒤에 rc라는 변수에 명령의 리턴 코드를 추가합니다.

명령이 0이 아닌 리턴 코드를 가지면 REXX는 오류 레이블로 분기합니다.

오류에 대한 신호는 QMF REXX 프로시저(ADDRESS QRW) 명령 환경으로부터 오류를 리턴시킵니다(REXX 호출 인터페이스 제외).

- 오류 레이블

오류에 대한 신호 명령어는 0이 아닌 리턴 코드가 발생하면 프로시저가 분기할 수 있는 레이블을 제공하도록 요구합니다. 레이블은 오류 처리 코드보다 선행합니다. 리턴 코드는 변수 rc에 있습니다. 다음과 같이 이 변수를 사용하여 다른 서브루틴으로 분기하거나 EXIT 명령어에 이 변수를 사용할 수 있습니다.

```
/* error handling code for a Procedure with logic */
error:
exit rc
```

REXX EXIT문에서 메시지 사용

앞의 절에서처럼 논리를 갖는 프로시저를 나가려면 REXX EXIT 명령을 사용할 수 있습니다. QMF는 항상 논리를 갖는 프로시저를 실행한 다음 메시지를 실행합니다. EXIT 명령어를 사용하는 경우, 사용자에게 표시되는 메시지는 다음 요인에 따라 달라집니다.

- 마지막 QMF 명령에 오류가 발생했는지 여부
- 리턴 코드가 0인지 여부

표2에서는 주어진 조건에 따라 사용자에게 보이는 메시지를 보여줍니다.

표 2. QMF에서 리턴된 메시지

코드	프로시저 리턴 코드	프로시저 완료시 메시지
No	0	프로시저는 실행됩니다.
No	0이 아님	프로시저로부터의 리턴 코드는 8입니다.
Yes	0	QMF가 제공하는 오류 메시지
Yes	0이 아님	QMF가 제공하는 오류 메시지

잘못된 QMF 명령과 0이 아닌 리턴 코드가 있으면 오류 메시지는 리턴 코드 메시지보다 우선합니다.

마지막 명령으로부터 오류 메시지를 표시하고 QMF 리턴 코드로 종료하려면, 다음 예제에서와 같이 MESSAGE 명령 및 EXIT DSQ_RETURN_CODE를 사용하십시오.

프로시저를 응용프로그램으로 사용

```
⋮  
"MESSAGE (TEXT='"dsq_message_text"'"  
exit dsq_return_code
```

그림 5. 오류 메시지 및 리턴 코드 표시

변수 `dsq_message_text` 및 `dsq_return_code`는 QMF에서 제공하는 REXX 변수입니다(이 변수의 자세한 목록은 241 페이지의 『REXX 언어 인터페이스』를 참조하십시오). 그림 6과 같이 후속 처리가 발생한 후 MESSAGE 명령 및 `dsq_message_text` 변수를 사용하여 메시지를 저장하고 표시할 수 있습니다.

```
/* Monthly report */  
Signal on error  
"DISPLAY TABLE JUNE_INFO"  
"PRINT REPORT"  
Exit(0);  
Error: Original_msg = dsq_message_text  
/* Saves error message. */  
"RUN PROC GENERAL_RECOVERY"  
/* This proc generates */  
/* new dsq_message_text. */  
"MESSAGE (TEXT='" Original_msg '"  
/* Display original error msg. */  
Exit dsq_return_code;
```

그림 6. 한 프로시저의 메시지 저장 및 검색

MESSAGE 명령에 관한 자세한 내용은 70 페이지의 『MESSAGE』를 참조하십시오.

논리를 갖는 프로시저로부터 REXX 프로그램 호출

응용프로그램을 호출하는 프로시저가 있을 수 있습니다. REXX 호출 인터페이스 응용프로그램을 논리를 갖는 프로시저로부터 호출할 때에는 응용프로그램에서 대체 변수에 지정한 앰퍼샌드(&)의 수에 주의해야 합니다. RUN QUERY WEEKLY_Q(&&DEPT=58에서처럼 호출중인 프로그램이 RUN 명령과 함께 대체 변수를 포함하고 있다면 더욱 그렇습니다.

대체 변수가 없는 REXX 프로그램 시작

REXX 프로그램에 대체 변수가 들어 있는 삽입 RUN 명령이 없으면, 다음 명령 중 하나를 사용하여 사용자 프로그램을 호출하십시오.

- ADDRESS 명령

이 명령어는 명령 환경을 설정합니다(명령 환경에 관한 자세한 내용은 281 페이지의 『부록D. ADDRESS QRW: QMF 명령 환경 사용』을 참조하십시오). 프로그램이 PANDA로 명명되고 CMS 환경 내에서 이를 호출하려면, 명령은 다음과 같습니다.

```
ADDRESS CMS "PANDA"
```

- CALL 명령

이 명령은 프로그램을 호출합니다. 프로그램 PANDA의 경우, 명령은 다음과 같습니다.

```
CALL PANDA
```

- 함수

다음과 같이 프로그램 PANDA를 함수로 호출할 수도 있습니다.

```
answer = PANDA()
```

이 명령에 관한 자세한 내용은 사용하는 시스템에 대한 REXX 참조 매뉴얼을 참조하십시오.

REXX 호출 중 하나를 사용하여 프로그램을 호출하려면 RUN 명령으로부터 대체 변수를 삭제하는 방안을 고려할 수도 있습니다. 이 경우 QMF는 변수에 대해 사용자에게 확인 메시지를 표시합니다.

대체 변수가 있는 REXX 프로그램 호출

REXX 응용프로그램이 대체 변수가 있는 QMF RUN 명령을 포함하는 경우 CMS program_name 또는 TSO program_name 중 하나를 사용하여 호출하십시오.

논리를 갖는 프로시저를 실행하건 논리를 갖는 프로시저에 의해 호출되는 호출 인터페이스 프로그램을 실행하건 명령은 같은 방법으로 QMF로 입력됩니다. 이러한 맥락에서, 호출 인터페이스 프로그램은 프로시저 자체의 논리 확장이 됩니다.

다음과 같은 명령을 고려하십시오.

프로시저를 응용프로그램으로 사용

```
RUN QUERY WEEKLY_Q (&DEPT=58
```

논리를 갖는 프로시저에서, 다음과 같이 대체 변수에 두 개의 앰퍼샌드를 사용하여 변수를 조회에 전달할 수 있습니다.

```
"RUN QUERY WEEKLY_Q (&&DEPT=58"
```

대체 변수에 하나의 앰퍼샌드만 있으면 QMF는 프로시저 자체에 대한 변수를 분석하며 변수를 조회로 전달할 수 없습니다.

REXX 호출 인터페이스 응용프로그램을 논리를 갖는 프로시저에서 호출하고 해당 응용프로그램이 RUN QUERY WEEKLY_Q(&DEPT=58 명령을 가지고 있는 경우 QMF는 호출중인 프로시저에 대해 수행했던 방식으로 변수를 분석합니다. 하나의 앰퍼샌드만 사용되었기 때문에 변수는 조회로 전달되지 않습니다.

논리를 갖는 프로시저에서 호출한 REXX 호출 인터페이스 응용프로그램에서 QMF로 변수를 전달하기 위해서는 다음과 같은 세 개의 선택항목이 있습니다.

- CMS 또는 TSO 명령을 사용하여 응용프로그램을 호출합니다.
응용프로그램을 호출하면 QMF는 어떠한 대체 변수도 처리하지 않습니다. 선행 명령에서 &DEPT=58은 조회로 전달되고 이 조회에서 대체 변수가 분석됩니다.
- 응용프로그램에서 모든 대체 변수를 논리를 갖는 프로시저에서 사용중인 것처럼 처리합니다.
앰퍼샌드를 모든 대체 변수에 추가하여 논리를 갖는 프로시저가 이를 분석하지 않도록 합니다.
- 글로벌 변수를 사용합니다.
응용프로그램의 시작 시점에서 글로벌 변수를 정의하고 QMF 세션 전체에서 이 변수를 사용할 수 있습니다.

제3장 호출 인터페이스

이 장은 QMF 호출 인터페이스에 대한 개요를 소개합니다. 특정 언어의 QMF 호출 인터페이스에 대한 특정 내용에 대해서는 해당 언어에 대한 부록A. 호출 인터페이스 언어의 샘플 코드 호출 인터페이스를 설명하는 절을 참조하십시오.

어셈블러

148 페이지의 『어셈블러 언어 인터페이스』

C 언어

171 페이지의 『C 언어 인터페이스』

COBOL

189 페이지의 『COBOL 언어 인터페이스』

FORTRAN

207 페이지의 『FORTRAN 언어 인터페이스』

PL/I 224 페이지의 『PL/I 언어 인터페이스』

REXX

241 페이지의 『REXX 언어 인터페이스』

호출 인터페이스란

프로그래밍 언어는 QMF 호출 인터페이스를 사용하여 QMF 명령을 실행할 수 있습니다. 모든 SAA 조회 명령은 호출 인터페이스를 통해 지원됩니다. QMF 호출 인터페이스는 다양한 프로그래밍 언어에 표준 인터페이스를 제공하고, 일반 기억 영역과 프로그램 변수에 대한 액세스를 제공합니다.

응용프로그램이 QMF 명령을 실행해야 할 경우, 우선 프로그램과 QMF 사이에 통신을 시작하기 위해 QMF 제공 루틴에 대한 호출을 실행해야 합니다. 이 호출은 QMF 제공 인터페이스 루틴으로 이루어집니다. QMF는 지원된 각 언어에 대한 루틴을 제공합니다.

호출 인터페이스

응용프로그램은 초기 시작 호출을 실행한 후 하나 이상의 QMF 명령을 발행할 수 있습니다. 응용프로그램은 각 QMF 명령을 발행하기 위해 QMF 제공 루틴을 호출합니다.

QMF 명령이 처리를 종료한 후에 QMF는 QMF의 상태를 나타내는 리턴 코드를 제공합니다. 호출 인터페이스는 명령 처리에 대한 다른 정보를 수집하고, QMF와 응용프로그램 둘다에서 액세스할 수 있는 변수에 이 정보를 저장합니다. 이 변수는 변수 풀이나 인터페이스 통신 영역에 포함되어 있습니다. 호출 인터페이스가 호출하는 응용프로그램으로 작업 상태가 리턴되면 응용프로그램은 이 변수를 참조할 수는 있지만 변수를 고칠 수는 없습니다.

응용프로그램이 더 이상 QMF를 사용하지 않으면 프로그램은 프로그램과 QMF간의 통신을 종료하는 호출을 발행합니다. 이 호출은 QMF 제공 루틴으로 이루어집니다.

QMF 호출 인터페이스를 사용하는 경우의 고려사항

- QMF 호출은 QMF가 QMF 명령 처리를 종료한 후에만 호출하는 응용프로그램으로 작업 상태를 리턴시킵니다.
- 호출이 처리되지 않을 때에는 QMF는 비활성 상태가 됩니다.
- 응용프로그램과 QMF는 변수 풀(pool)이나 인터페이스 통신 영역에 저장된 리턴 코드와 변수 데이터를 가지고 통신합니다.
- 모든 QMF 명령은 대문자 영문자로 코드화됩니다.
QMF 자국어 피쳐(NLF)를 사용할 경우, QMF 명령은 기본 언어로 지정된 NLF 언어로 작성되며 대문자로(또는 폴드로) 작성됩니다.
- 전달되는 명령의 최대 길이는 256바이트입니다.

23 페이지의 그림7에서는 응용프로그램이 호출 인터페이스를 통해 QMF로 명령을 전달하는 방법을 보여줍니다.

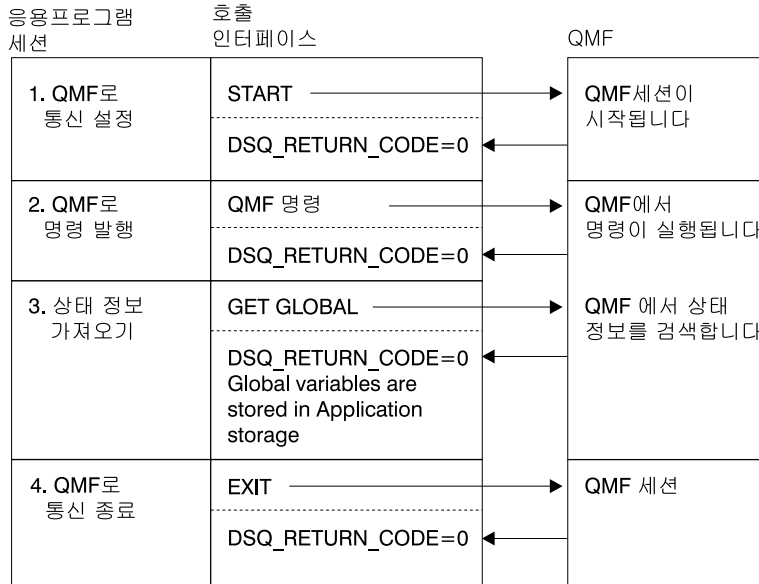


그림 7. 응용프로그램은 QMF 호출 인터페이스를 사용하여 QMF와 통신합니다.

호출 인터페이스를 통해 발행한 명령의 결과는 보통 대화식으로 명령을 발행한 경우와 같습니다.

인터페이스 통신 영역의 정의(DSQCOMM)

QMF는 지원되는 각 프로그래밍 언어에 대해 인터페이스 통신 매크로를 제공합니다. 이 매크로는 다음 정보를 포함합니다.

- 인터페이스 통신 영역(DSQCOMM) 또는 통신 변수
- 리턴 코드 및 이유 코드 정의
- QMF에 대한 함수 호출의 정의

이 매크로는 앞의 목록에서 설명된 변수를 포함하는 몇몇 기억영역을 정의합니다. 이 기억영역은 호출 인터페이스 통신 영역이고 이 영역에 저장된 변수는 QMF만이 값을 바꾼다고 해도 QMF와 호출 인터페이스 응용프로그램 모두에서 사용할 수 있습니다. 응용프로그램은 이들 변수를 읽기 전용으로만 볼 수 있습니다.

REXX 호출 인터페이스는 통신 영역을 사용하지 않고 QMF에서 제공하는 인터페이스 통신 변수를 사용합니다.

호출 인터페이스

QMF 호출 인터페이스 통신 영역이 모든 호출 인터페이스 호출 때마다 필요합니다. 호출 인터페이스 통신 영역의 기억영역은 QMF를 사용하는 프로그램에 의해 할당됩니다.

START 명령은 QMF 세션의 고유한 인스턴스 또는 발생을 설정합니다. START 명령은 다음 위치에 하나의 QMF 세션만 설정할 수 있습니다.

- TSO 주소 공간
- 단일 CMS 가상 기계
- 단일 CICS 트랜잭션으로부터

START 명령을 실행할 경우 QMF는 인터페이스 통신 영역 또는 변수를 갱신합니다.

다음 예외를 제외하고 응용프로그램은 인터페이스 통신 영역 또는 변수를 바꿀 수 없습니다.

DSQ_COMM_LEVEL

DSQ_COMM_LEVEL을 DSQ_CURRENT_COMM_LEVEL의 값에 설정하여 DSQCOMM의 레벨을 확인합니다. 이것은 REXX에는 적용되지 않습니다.

DSQ_INSTANCE_ID

QMF에서 호출 인터페이스 프로그램을 호출하려면 QMF가 초기 START 명령에 의해 설정된 값에 변수를 다시 설정할 수 있도록 처음 호출시 DSQ_INSTANCE_ID를 0에 설정해야 합니다.

START 명령 뒤의 모든 호출은 QMF 인스턴스에 상응하는 인터페이스 통신 영역의 주소를 제공해야 합니다. 응용프로그램은 정확한 인터페이스 통신 영역을 지적해야 합니다.

지원된 각 언어는 인터페이스 통신 영역을 설명하는 고유한 통신 매크로를 가지고 있습니다. 응용프로그램을 이동시키려면 변수 값 대신 변수 이름에 따라 변수를 언급해야 합니다. 다른 시스템에서는 변수 값이 다를 수 있기 때문입니다.

인터페이스 통신 영역 또는 변수는 25 페이지의 표3의 정보를 포함하는데, 이 정보는 프로그램 호출에 의해 변경될 수 없습니다.

표 3. 변경할 수 없는 DSQCOMM 필드

필드	설명
리턴 코드	QMF가 명령을 처리한 뒤에 QMF 처리 상태를 나타냅니다.
인스턴스 ID	START 명령에 의해 시작된 QMF의 인스턴스를 확인합니다.
완료 메시지 ID	<p>명령이 발행되었다면 QMF가 사용자의 터미널에 표시한 메시지의 메시지 ID를 포함합니다.</p> <p>이 필드는 모든 QMF 명령이 완료될 때 설정됩니다. 이 필드는 QMF가 명령의 마지막 부분에 표시한 메시지를 포함합니다.</p>
조회 메시지 ID	<p>RUN QUERY 명령의 결과인 QMF 메시지의 메시지 ID를 포함합니다. 이것은 조회에 표시되는 메시지의 메시지 ID입니다.</p> <p>이 필드는 조회가 실행되는 동안 오류가 발생한 경우에 설정됩니다. 이 필드는 QMF가 명령의 마지막 부분에 조회 오브젝트 범위에 표시하는 메시지를 포함합니다.</p>
오류 상태인 START 명령 매개 변수	매개변수 오류로 인해 START 명령이 실패할 경우 오류 상태에 있는 매개변수 수를 포함합니다.
취소 표시기	QMF가 명령을 실행하는 동안 사용자가 명령 처리를 취소했는지 나타냅니다.
완료 메시지	QMF가 사용자의 터미널에 표시하는 완료 메시지를 포함합니다.
조회 메시지	<p>RUN QUERY 명령의 결과인 조회 메시지 텍스트를 포함합니다. 이것은 QMF가 사용자의 조회에 표시하는 텍스트입니다.</p> <p>예를 들어, 오류가 있는 조회 오브젝트를 실행하는 경우 QMF는 조회의 실행을 막는 오류를 설명하는 메시지를 표시합니다. 그러면 조회 메시지에는 이 오류 메시지 텍스트가 포함됩니다.</p>

리턴 코드

리턴 코드는 각 QMF 호출 인터페이스를 호출한 뒤에 리턴됩니다. 리턴 코드 값은 QMF와 함께 제공되는 통신 매크로에 의해 설명됩니다.

응용프로그램을 시스템간에 이동시키려면 변수 이름별로 이들 코드의 값을 참조해야 합니다. 다른 시스템에서는 변수 값이 달라질 수 있기 때문입니다.

호출 인터페이스로부터의 리턴 코드는 다음 조건을 나타냅니다.

- QMF는 성공적으로 요청을 처리합니다.
- QMF는 경고 조건에도 불구하고 요청을 처리합니다.

호출 인터페이스

- QMF는 명령을 제대로 처리하지 않습니다.
- 심각한 오류로 인해 QMF의 이 인스턴스는 종료합니다.

각 리턴 코드의 정의에 대해서는 이 안내서의 적절한 프로그래밍 언어 절을 참조하십시오.

호출 인터페이스를 사용하는 경우의 명령

호출 인터페이스를 사용하여 프로시저에서 사용한 QMF 명령을 사용할 수 있습니다. 그러나 다음과 같이 호출 인터페이스에 대해 특수 구문을 가지는 세 개의 명령이 있습니다.

- START
- GET GLOBAL, 확장 구문
- SET GLOBAL, 확장 구문

START는 호출 인터페이스에서만 작동합니다. REXX 이외의 다른 언어로 작성된 호출 인터페이스 응용프로그램에서 GET GLOBAL 및 SET GLOBAL을 사용하려면 확장 구문을 사용하십시오. SET GLOBAL 명령의 확장 구문을 통해 최대 32 768자 길이의 글로벌 변수를 설정할 수 있습니다. 응용프로그램에서 GET GLOBAL 및 SET GLOBAL 명령 사용에 관한 자세한 내용은 65 페이지의 『GET GLOBAL』 및 74 페이지의 『SET GLOBAL: 확장 구문』을 참고하십시오.

호출 인터페이스 응용프로그램에서 사용할 수 있는 GET GLOBAL, SET GLOBAL 명령과 다른 명령에 대해서는 59 페이지의 『제7장 응용프로그램의 QMF 명령』을 참조하십시오. 각 언어에 대한 START 및 SET GLOBAL 명령의 예제를 보려면 각 언어에 대한 샘플 프로그램을 참조하십시오.

어셈블러

152 페이지의 『어셈블러 프로그래밍 예제』

C 언어

176 페이지의 『C 언어 프로그래밍 예제』

COBOL

194 페이지의 『COBOL 프로그래밍 예제』

FORTRAN

210 페이지의 『FORTRAN 프로그래밍 예제』

PL/I 227 페이지의 『PL/I 프로그래밍 예제』

REXX

245 페이지의 『REXX 프로그래밍 예제』

응용프로그램에서 QMF 시작

응용프로그램으로부터 어떠한 명령이라도 실행하기 전에 먼저 QMF를 시작해야 합니다. 호출 인터페이스를 사용할 경우 START 명령을 발행하여 QMF를 시작하십시오. 한 번에 하나의 QMF 세션만 가질 수 있습니다.

응용프로그램은 START 명령을 발행하여 QMF가 이미 시작되었는지 테스트할 수 있습니다. QMF가 이전에 시작되지 않았으면 지금 시작됩니다. QMF가 이전에 시작되었으면 리턴 코드는 0이 아니며 다음 메시지 번호와 메시지가 수신됩니다.

DSQ50719 QMF는 이미 사용중입니다. 2차 세션은 허용되지 않습니다.

REXX 호출 인터페이스로 다음 프로그램도 실행할 수 있습니다.

```
/* test to see if QMF is active */
"SUBCOM QRW"
if rc = 0
  then say "QMF is active"
else say "QMF is not active"
```

START 명령이 심각하지 않은(리턴 코드 4 또는 8) 오류를 발생시키면 QMF는 오류 상태에서 그대로 시작됩니다. 이 경우에 EXIT 명령을 발행하여 QMF를 중단시킬 수 있습니다. START 명령을 다시 발행할 수도 있습니다. 오류가 지속되면 인터페이스 통신 영역이나 QMF 추적 데이터 출력에서 오류의 원인이 검사됩니다.

QMF에 매개변수를 전달하려면 START 명령에 원하는 명령 키워드를 지정하십시오.

시작 명령과 함께 사용되는 구문과 키워드에 대한 자세한 내용은 77 페이지의 『START』를 참조하십시오.

호출 인터페이스 응용프로그램 실행

호출 인터페이스 응용프로그램을 실행하려면 대화식 QMF를 실행하는 것처럼 실행 환경을 설정해야 합니다.

호출 인터페이스

환경을 설정하고, 호출 인터페이스 응용프로그램을 컴파일하고, 실행하는 데 대한 내용은 147 페이지의 『부록A. 호출 인터페이스 언어의 샘플 코드』에서 사용하는 언어에 맞는 적절한 코딩 샘플을 참조하십시오.

QMF에서 호출 인터페이스 사용

CICS 사용자를 위한 주의사항

CICS 환경에서는 QMF로부터 호출 인터페이스를 사용할 수 없습니다.

CICS를 제외한 QMF에 지원되는 모든 환경에서는 QMF에서 호출 인터페이스를 사용하여 QMF 임시 기억영역을 변경하는 응용프로그램을 실행할 수 있습니다. 예를 들어, QMF 세션중에 호출 인터페이스를 통해 파일을 내보내거나 가져올 수 있습니다.

CMS 또는 TSO 명령을 사용하여 응용프로그램을 호출할 수 있습니다. 응용프로그램으로부터 유효한 QMF 명령을 실행할 수 있습니다. QMF가 이미 사용중이므로 START 명령을 발행하면 안 됩니다.

먼저 QMF를 호출하기 전에 DSQCOMM 인스턴스 ID(DSQ_INSTANCE_ID)를 0으로 설정하십시오. QMF는 현재 인스턴스를 결정하고 이후의 QMF 호출에 사용하기 위해 DSQ_INSTANCE_ID를 갱신합니다.

오류 처리

CICS 환경에서 QMF를 실행하지 않는 경우 QMF에 제공된 REXX 변수 또는 DSQCOMM 통신 영역의 유사한 값을 사용하여 응용프로그램 오류를 처리할 수 있습니다.

예를 들어, REXX 변수 dsq_message_text 또는 DSQCOMM의 메시지 텍스트 필드는 QMF 메시지를 포함합니다.

REXX에서, QMF는 모든 QMF 명령이 완료될 때 다음 값 중 하나를 변수 dsq_return_code에 지정합니다.

dsq_success

성공적인 명령의 완료

dsq_warning

경고 메시지를 나타내는 정상적 완료

dsq_failure

명령은 제대로 실행되지 않음

dsq_severe

심각한 오류. QMF 세션이 종료됨

REXX 이외의 다른 언어의 경우, QMF는 DSQCOMM의 리턴 코드 필드 DSQ_RETURN_CODE에 같은 값을 표시합니다.

응용프로그램에서 이 리턴 코드와 값을 사용할 수 있습니다. 다음 예제에서는 REXX 호출 인터페이스 응용프로그램에서 오류 처리 변수를 사용하는 방법을 보여줍니다.

```

:
call dsqcix "CONVERT QUERY MYQUERY"
if dsq_return_code dsq_success then ...
:
call dsqcix "PRINT REPORT"
if dsq_return_code=dsq_severe | dsq_return_code=dsq_failure then ...

```

QMF는 메시지 번호와 메시지 텍스트를 포함하는 변수도 제공합니다.

각 DSQCOMM의 변수 또는 필드에 대한 목록을 보려면 147 페이지의 『부록A. 호출 인터페이스 언어의 샘플 코드』에서 각 언어에 대한 적절한 절을 참조하십시오.

CICS에서 호출 인터페이스 프로그램 실행

QMF 호출 인터페이스를 사용하는 프로그램을 실행하려면 CICS 프로그램을 설치하는 일반적인 방법을 사용하여 이들 프로그램을 CICS로 설치합니다. CICS의 응용프로그램에 대해서는 *CICS for VSE/ESA Application Programming Guide*를 참조하십시오. QMF 응용프로그램 설치에 대해서는 *CICS for VSE/ESA System Definition Guide*를 참조하십시오.

정상적인 CICS 필요조건 이외에, 다음 고려사항이 CICS에서 실행되는 모든 QMF 호출 인터페이스 프로그램에 적용됩니다.

- 환경

프로그램이 QMF 제품을 호출할 때 프로그램은 대화식 QMF 제품과 같은 특성을 나타냅니다. 이 프로그램은 매우 큰 대화식 프로그램이 됩니다.

QMF는 CICS 명령을 포함하는 어셈블러 언어 프로그램입니다. 다른 어셈블러 언어 프로그램이나 고급 언어(VS COBOL II, PL/I 또는 C/370™) 프로그램 중 하나와 링크될 수 있습니다. 상위 레벨 언어를 사용하여 QMF를 호출하려면 상위 레벨 언어 프로그램이 먼저 연결되어야 하고 자원 정의 온라인(RDO) 프로그램 정의에서 고급 레벨 언어가 지정되어야 합니다. 각 고급 프로그램에는 특정 CICS 고려사항 및 제한사항이 적용됩니다. 고급 언어 프로그래밍 안내서 및 *CICS Application Programming Guide*를 참조하십시오.

CICS에서 기본 QMF 시작 매개변수를 덮어쓰려면 START 명령에 이 키워드를 지정하십시오. 예를 들어 호출 인터페이스의 기본 모드는 BATCH입니다. 대화식 QMF 세션을 실행하려면 DSQSMODE=INTERACTIVE를 사용하여 START 명령을 발행해야 합니다.

- 프로그램 실행 레벨

QMF 버전 3 릴리스 1 수정판 1의 경우, QMF 제품 인터페이스와 주 QMF 프로그램간의 인터페이스는 사용자의 응용프로그램보다 낮은 프로그램 레벨에서 실행되도록 변경됩니다. 이 변경으로 인해 사용자 프로그램은 QMF에 의해 설정된 처리 조건과 같은 환경적인 조건에 의해 영향받지 않습니다.

CICS/OS/390 사용자를 위한 주의사항

3.1로 마이그레이트한 후 QMF 6 호출 인터페이스를 사용하려면 현재 QMF 호출 인터페이스를 사용하는 프로그램을 링크 편집해야 합니다. 후속 QMF 릴리스에서 마이그레이트중이면 다시 링크 편집할 필요가 없습니다.

- CICS 영역(OS/390) 또는 분할 영역(VSE) 고려사항
 QMF 인터페이스 통신 모듈과 주 QMF 모듈을 가지고 있는 사용자 프로그램은 같은 영역이나 분할 영역에서 실행되어야 합니다. QMF 설치중에 설명된 것처럼 QMF 자원은 QMF를 실행하는 CICS 영역이나 분할 영역에도 할당되어야 합니다.
- 데이터베이스
 - **VSE 또는 VM용 DB2:** 호출 인터페이스를 통해 QMF를 호출할 경우 CICS 트랜잭션은 이미 설치된 데이터베이스 패키지를 사용하여 QMF를 실행하며 다른 조치는 필요하지 않습니다.
 - **OS/390용 DB2 UDB:** 프로그램을 호출하는 CICS 트랜잭션 역시 자원 제어 표(RCT) 엔트리로 DB2에 설명되어야 합니다. RCT 엔트리에 대해서는 *DB2 UDB for OS390 Administration Guide* 및 *CICS System Definition Guide*를 참조하십시오.
 RCT PLAN 이름은 호출 인터페이스 프로그램과 QMF 제품 모두에 대해 같아야 합니다.

제4장 응용프로그램에 명령 인터페이스 사용

QMF는 ISPF 대화 상자로부터 QMF 서비스를 사용하기 위한 응용프로그램 인터페이스를 제공합니다. 이 인터페이스는 명령 인터페이스입니다. 명령 인터페이스는 QMF에서 실행되는 ISPF 대화 상자로부터 QMF 명령을 발행할 수 있게 해 줍니다. 이 인터페이스를 사용하여 그림8과 같이 QMF는 ISPF 변수 풀을 통해 대화와 통신합니다.

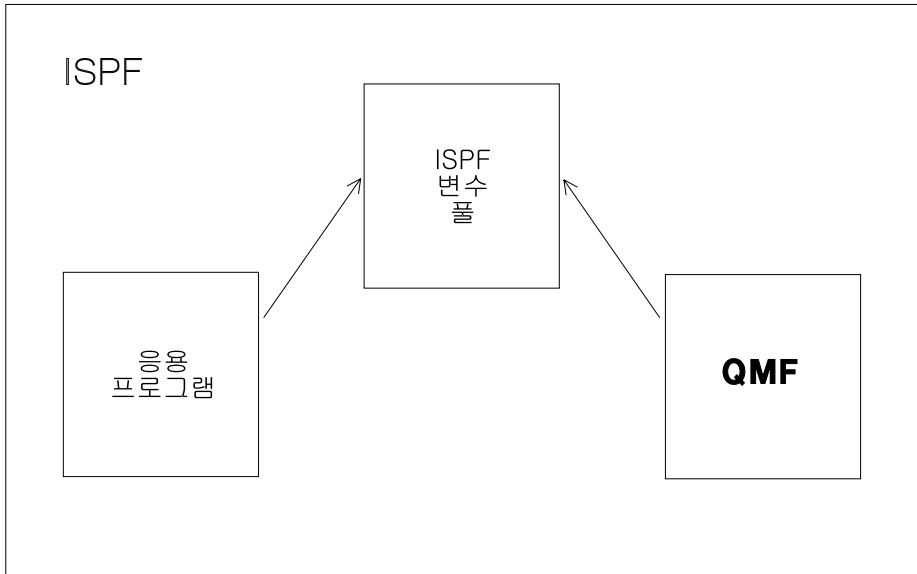


그림 8. QMF와 대화하는 QMF 명령 인터페이스 응용프로그램

CICS 사용자를 위한 주의사항

QMF 명령 인터페이스가 실행되려면 ISPF가 필요하지만 ISPF는 CICS 환경에서는 실행되지 않습니다. 그러므로 CICS에서 응용프로그램을 개발하기 위해서는 QMF 호출 인터페이스를 사용해야 합니다.

응용프로그램에 명령 인터페이스 사용

효과적으로 명령 인터페이스를 사용하려면 ISPF 서비스와 변수 풀을 이해해야 합니다. ISPF 사용에 대한 자세한 내용은 *ISPF: Dialog Management Guide and Reference*를 참조하십시오.

명령 인터페이스(DSQCCI)를 사용하기 위해서는 QMF 세션을 실행해야 합니다. 명령 인터페이스를 사용하여 QMF 세션을 시작할 수 없습니다. ISPSTART 명령을 사용하여 ISPF에서 QMF 세션을 시작할 수 있습니다.

```
ISPSTART PGM(DSQMFE) NEWAPPL(DSQE) PARM(...)
```

ISPF SELECT PGM 서비스를 사용하여 QMF 명령 인터페이스 프로그램 DSQCCI를 호출하여 ISPF 대화에서 QMF 명령을 사용할 수 있습니다. SELECT PGM 서비스의 PARM 옵션을 사용하여 QMF 명령을 제공해야 합니다. 명령 인터페이스를 사용하는 프로그램을 실행하려면, 다음 단계를 따라야 합니다.

1. ISPF를 시작합니다.
2. QMF를 시작합니다.
3. CMS 또는 TSO 명령을 사용하여 프로그램을 실행합니다.

중요: 이 단계 중 하나라도 생략되면 프로그램은 실행되지 않습니다.

명령 인터페이스를 사용하는 프로그램 작성: 예제

명령 인터페이스를 사용하여 조회 이름을 지정하고, 조회를 실행하고, 보고서를 표시하는 프롬프트를 사용자에게 제시하는 ISPF 패널을 표시하려 한다고 가정해 보십시오.

이 시나리오의 경우, 다음을 수행하십시오.

1. 명령 인터페이스 REXX 프로그램을 작성합니다. 프로그램에서는 다음을 수행합니다.

- a. DISPLAY 서비스를 사용하여 ISPF 패널 QRYNAME을 표시합니다.

```
ADDRESS ISPEXEC "DISPLAY PANEL(QRYNAME)"
```

- b. 이전 DISPLAY 서비스에서 사용자가 입력한 내용에 따라 QMF 조회를 실행합니다. 여기에서 ISPF 변수 QNAME에는 QMF 조회의 이름이 포함됩니다.

```
ADDRESS ISPEXEC "SELECT PGM(DSQCCI) PARM(RUN QUERY" QNAME ")"
```

c. 다음 명령을 사용하여 사용자는 조회의 결과를 볼 수 있습니다.

```
ADDRESS ISPEXEC "SELECT PGM(DSQCCI) PARM(INTERACT)"
```

2. ISPF를 시작합니다.
3. QMF를 시작합니다.
4. QMF 명령행에서 CMS 또는 TSO 명령을 사용하여 프로그램을 호출합니다. 예를 들어, 프로그램 이름이 GETINFO인 경우 명령은 사용자 시스템에 따라 다음 중 하나로 표시될 수 있습니다.

```
CMS GETINFO
TSO GETINFO
```

명령 인터페이스 호출

명령 인터페이스는 DSQCCI로 명명된 프로그램입니다. ISPF SELECT 서비스를 통해 프로그램에서 이를 호출할 수 있습니다.

ISPF SELECT 서비스를 통해 명령 인터페이스를 호출할 때에는 PARM 피연산자에서 실행될 대문자 QMF 명령을 제공하십시오. 다음 명령을 발행하십시오.

```
SELECT PGM(DSQCCI) PARM(qmf_command)
```

QMF 프로파일 설정에 관계없이 매개변수로 지정된 모든 QMF 명령은 대문자여야 합니다. ISPF는 자동으로 명령을 소문자에서 대문자로 전환하지는 않습니다. 소문자로 QMF 명령을 지정하면 QMF는 그것을 인식하지 않습니다. QMF에서 QMF 명령을 실행하는 동안 프롬프트가 실행되게 하려면 QMF 명령 바로 앞에 명령을 코드화하십시오. INTERACT 명령에 대한 자세한 내용은 66 페이지의 『INTERACT』를 참조하십시오.

호출시 NEWPOOL 또는 NEWAPPL 옵션을 지정하지 마십시오. NEWPOOL 또는 NEWAPPL 옵션을 생략하면 명령 인터페이스가 응용프로그램 변수를 액세스할 수 있게 됩니다. 명령 인터페이스는 QMF와 응용프로그램간을 통신하기 위해 공유 풀을 사용합니다.

SELECT 서비스의 경우 RUN QUERY 명령에 이중 앰퍼샌드 기호를 사용해야 합니다. 이렇게 하면 ISPF는 해당 변수를 자신의 변수로 해석하지 못하게 됩니다.

END 명령

명령 인터페이스(DSQCCI)가 실행되는 동안 일반 사용자가 END 명령을 발행하면 DSQCCI 호출이 종료되고 호출하는 응용프로그램으로 제어를 리턴시킵니다. QMF 세션은 사용중입니다. 이상 종료(비정상 종료)만이 명령 인터페이스 호출중에 QMF 세션을 종료합니다.

명령 인터페이스를 호출하는 동안 EXIT 명령이나 심각한 오류가 발생하면 QMF는 세션 종료를 표시하도록 DSQCSESC를 설정합니다. DSQCCI를 호출할 프로그램이 종료되고 QMF로 제어를 리턴시킨 후에 QMF 세션이 종료됩니다.

명령 인터페이스에서 변수 사용

STATE 명령은 각 QMF 제공 변수에 대해 현재의 값을 제공합니다. 명령은 명령 인터페이스에서만 사용될 수 있습니다. 이 명령이 발행되면 VPUT 명령을 통해 ISPF 변수 풀에 QMF 변수를 추가할 수 있습니다. 37 페이지의 표4에서는 QMF가 ISPF 변수 풀에 추가한 사용할 수 있는 QMF 변수의 서브세트를 보여줍니다.

표 4. ISPF 변수 풀에 있는 QMF 변수

변수 유형	변수 이름	설명	
STATE 명령	DSQAAUTH	DSQAPLEN	응용프로그램이 STATE 명령을 발행한 경우 QMF는 이 변수를 갱신합니다.
	DSQABATC	DSQAPLNG	
	DSQACMDM	DSQAPPFK	
	DSQACRSR	DSQAPPRT	
	DSQADBCS	DSQAPRMP	
	DSQADBMG	DSQAPSPC	
	DSQAIACT	DSQAPSYN	
	DSQAITEM	DSQAPTRC	
	DSQAITLO	DSQAPWID	
	DSQAITMN	DSQAQMF	
	DSQAITMO	DSQAREVN	
	DSQALANG	DSQAROWS	
	DSQAMODL	DSQASUBI	
	DSQAMODP	DSQASUBP	
	DSQAOGRP	DSQATRAC	
	DSQAPCAS	DSQAVARN	
	DSQAPDEC		
CONVERT 명령	DSQCL nmn	QMF는 CONVERT 명령을 처리할 때 이 변수를 갱신합니다.	
	DSQCQ nmn		
	DSQCQCNT		
	DSQCQLNG		
	DSQCQTYP		

표 4. ISPF 변수 풀에 있는 QMF 변수 (계속)

변수 유형	변수 이름	설명
명령 메시지	DSQCATTN	명령 인터페이스에 의해 발행되는 명령을 처리할 때마다 QMF는 이 변수를 갱신합니다.
	DSQCIM00	
	DSQCIMnn	
	DSQCIMID	
	DSQCIMNO	
	DSQCIMSG	
	DSQCSESC	
조회 메시지	DSQCIQ00	QMF는 RUN QUERY가 오류 메시지를 리턴시킬 때 이 변수를 갱신합니다.
	DSQCIQnn	
	DSQCIQID	
	DSQCIQMG	
	DSQCIQNO	
	DSQCISQL	

ISPF 변수 풀에 있는 QMF 변수를 사용하려면 변수에 대해 8자의 이름을 사용하십시오. 값에 대한 설명과 이 변수의 확장 이름에 대해서는 285 페이지의 『부록 F. QMF 글로벌 변수 표』를 참조하십시오.

명령 인터페이스 리턴 코드

명령 인터페이스에 대한 리턴 코드는 응용프로그램의 언어와 관계없이 동일합니다. 리턴 코드는 양수이거나 0이 될 수 있습니다. 0 값은 성공적인 실행을 나타냅니다. 양수 값은 실행이 실패했거나 비정상적인 상태를 나타냅니다.

리턴 코드는 사용자의 EXEC 또는 CLIST에 있는 변수에 나타납니다. REXX EXEC를 실행하면 리턴 코드는 RC라는 REXX 변수에 있으며, CLIST를 실행하면 리턴 코드는 CLIST 변수 &LASTCC에 있습니다.

다음 예제에서는 리턴 코드를 검사하는 EXEC를 보여줍니다.

예제

응용프로그램에는 다음과 같은 코드가 들어 있습니다.

```
ADDRESS ISPEXEC SELECT PGM(DSQCCI) PARM(RUN QUERYA (FORM=FORMA))
  Select
  When (RC = 0) Then nop
  When (RC = 64) Then
    Say "You must run QMF with ISPF to use command interface."
  When (RC = 100) Then
    Say "You need to start QMF before you begin your application"
  Otherwise
    Say "Unexpected error ("RC") from QMF command interface."
End
```

코드는 조회를 실행한 후에 REXX RC를 사용하여 오류가 있는지 테스트합니다.

EXEC 또는 CLIST에서처럼 프로그램 모듈의 오류를 처리하는 코드도 추가할 수 있습니다.

0에서 16까지의 리턴 코드

0에서 16까지의 리턴 코드는 명령 인터페이스와 함께 전달된 명령의 QMF 처리 방식에 대해 설명합니다. 명령 인터페이스가 이 코드 중 하나를 리턴시킬 때에는 응용프로그램의 ISPF 공유 풀에 있는 QMF 명령 메시지 변수 값도 함께 리턴시킵니다. 코드는 표5와 같습니다.

표 5. 0에서 16까지의 리턴 코드

값	설명
0	성공적인 실행
4	QMF 세션은 EXIT 또는 END 명령으로 종료 표시됩니다.
8	실행이 실패해도 오류는 세션 종료를 표시하지 않습니다.
16	심각한 오류: 세션 종료가 표시되었습니다.

리턴 코드 4는 세션 종료를 표시한 명령에서만 발생합니다. 응용프로그램이 다른 명령을 실행하려 하면 QMF는 사용자에게 다른 리턴 코드 값을 리턴시킵니다.

20 이상의 리턴 코드

이 코드는 일반적으로 명령 인터페이스(DSQCCI)에서 실패가 발생했음을 나타냅니다. 오류로 인해 변수가 응용프로그램의 공유 풀에 복사되지 않게 됩니다. 결과적으로, QMF 변수는 유효하지 않게 되거나 설정되지 않습니다. 프로그램이 STATE 명령을 사용하면 STATE 변수에도 똑같은 사항이 적용됩니다(변수는 응용프로그램의 공유 풀에 복사되면 『설정』된 것입니다).

이 리턴 코드는 일반적으로 0에서 16까지 범위의 오류보다 더 심각한 오류를 나타냅니다. 경우에 따라 IBM 고객 지원팀에 서비스를 요구할 수 있습니다.

다음 표에서는 20 이상의 값을 가진 리턴 코드를 설명합니다. 공유 변수는 QMF 변수를 나타냅니다(현재 명령이 STATE 명령이면 STATE 변수를 나타냅니다).

어떤 코드의 경우, 명령은 실행되지만 공유 변수는 설정되지 않습니다. 명령이 STATE 명령이라면 문제가 있는 것처럼 생각될 것입니다. 이것은 QMF가 STATE 명령을 적절하게 실행했음을 의미합니다. QMF는 명령 인터페이스가 갱신된 공유 QMF 및 STATE 변수를 설정할 것을 기대했지만 오류 코드의 설명에 나와 있는 오류로 인해 명령 인터페이스는 이 작업을 수행하지 못했습니다. 코드는 표6과 같습니다.

표 6. 20 이상의 리턴 코드

값	설명
20	사용자 종료 루틴이 명령 인터페이스를 호출했습니다. 이 호출은 항상 유효하지 않은 것은 아닙니다. 명령 인터페이스에 제공된 명령은 실행되지 않았습니다. 공유 변수는 설정되지 않았습니다.
24	ISPF VCOPY 명령에서 오류가 발생했습니다. 명령 인터페이스에 제공된 명령은 실행되었습니다. 공유 변수는 설정되지 않았습니다.
32	ISPF VREPLACE 명령에서 오류가 발생했습니다. 명령 인터페이스에 제공된 명령은 실행되었습니다. 공유 변수는 설정되지 않았습니다.
36	ISPF VPUT 명령에서 오류가 발생했습니다. 명령 인터페이스에 제공된 명령은 실행되었습니다. 공유 변수는 설정되지 않았습니다.
40	ISPF VREPLACE 명령에서 오류가 발생했습니다. 이 코드는 STATE 명령의 실행에만 적용됩니다. 명령 인터페이스에 제공된 명령은 『실행되었으나』 공유 변수는 설정되지 않았습니다.
44	ISPF VPUT 명령에서 오류가 발생했습니다. 이 코드는 STATE 명령의 실행에만 적용됩니다. QMF 변수는 설정되었으나 STATE 변수는 설정되지 않았습니다.
60	명령 인터페이스에 대한 유효하지 않은 호출. 예를 들어, 사용자가 QMF 프롬프트 패널에서 응용프로그램을 호출했으며 응용프로그램은 명령 인터페이스를 호출했습니다. 명령 인터페이스에 제공된 명령은 실행되지 않았습니다. 공유 변수는 설정되지 않았습니다.
64	ISPF 환경에서는 호출되지 않습니다. DSQCCI가 실행중이고 ISPF가 비활성 상태이면 이 오류가 발생합니다. 예를 들어, 사용자는 DSQCCI를 ISPF SELECT PGM 명령을 사용하지 않고 호출했을 수 있습니다.
100	기준을 찾는 데 실패했습니다. 이 오류는 응용프로그램이 QMF 실행되지 않은 상태에서 QMF 명령을 발행할 때 발생합니다. 응용프로그램을 시작하기 전에 QMF를 시작해야 합니다. 명령 인터페이스에 전달된 명령은 실행되지 않았습니다. 공유 변수는 설정되지 않았습니다.
104	기준을 찾는 데 실패했습니다. 명령 인터페이스에 전달된 명령은 실행되지 않았습니다. 공유 변수는 설정되었으나 유효하지 않습니다.

제5장 ISPF를 사용하는 QMF 응용프로그램 작성

자체의 인터페이스를 가지며 모든 QMF 패널을 무시하는 응용프로그램을 작성할 수 있습니다. 이 응용프로그램을 작성하는 한 가지 방법은 ISPF를 사용하여 자체의 패널을 작성하고 사용자의 입력항목을 변수로서 QMF에 전달하는 것입니다. 다른 ISPF 서비스를 활용하며 QMF 오브젝트를 작성하거나 읽을 수 있습니다.

CICS 사용자를 위한 주의사항

ISPF는 CICS 환경에서 실행되지 않으므로 ISPF 서비스도 CICS 환경에서 사용할 수 없습니다.

ISPF는 메인프레임 시스템에서 일반 사용자 인터페이스를 제공하도록 도와줍니다. QMF 호출 인터페이스 또는 명령 인터페이스와 함께 ISPF를 사용할 수 있습니다.

이 장은 ISPF와 함께 호출 인터페이스를 사용하는 경우의 고려사항에 대해 간단하게 설명합니다. 호출 인터페이스 사용에 대한 일반 정보는 21 페이지의 『제3장 호출 인터페이스』를 참조하십시오. 명령 인터페이스 사용에 대한 정보는 33 페이지의 『제4장 응용프로그램에 명령 인터페이스 사용』을 참조하십시오.

ISPF 응용프로그램에서 QMF의 시작 및 실행

호출 인터페이스는 다른 프로그램과 작업할 때처럼 ISPF와 작업합니다. 그러나 몇 가지 고려사항이 있습니다.

호출 인터페이스는 **ISPF** 대화의 언어와 일치해야 합니다.

예를 들어 ISPF 대화가 PL/I 프로그램이면 PL/I에 대해 QMF 호출 인터페이스를 사용해야 합니다.

올바른 언어 **ID**를 사용해야 합니다.

ISPF를 사용하는 QMF 응용프로그램 작성

DSQn의 ID를 사용하여 ISPF 응용프로그램을 시작해야 합니다. 여기에서 *n*은 자국어 피처(NLF) ID입니다. 이 응용프로그램 ID는 QMF가 ISPF 환경을 겹쳐쓰지 못하게 합니다(예: 기능 키 설정 및 레이블). QMF를 시작하는 응용프로그램을 시작하려면, 다음 ISPF 문을 사용하십시오.

```
SELECT PGM(MYPROG) NEWAPPL(DSQn)
```

여기서 *n*은 NLF ID입니다. 그런 다음 PL/I 프로그램 MYPROG는 호출 인터페이스 START 명령을 사용하는 QMF를 시작합니다.

ID DSQn은 QMF가 시작된 후에도 ISPF 환경이 그대로 남아 있도록 합니다.

NLF ID 목록에 대해서는 80 페이지의 표7을 참조하십시오.

STATE 명령 대신 GET GLOBAL 또는 SET GLOBAL을 사용합니다.

GET GLOBAL 및 SET GLOBAL 명령은 모든 QMF 글로벌 변수에 대해 작동됩니다. STATE 명령은 상태 정보를 포함하는 변수에 대해서만 작동됩니다. 285 페이지의 『부록F. QMF 글로벌 변수 표』에서 이 변수의 표를 참조하십시오.

변수를 포함하는 조회 실행

응용프로그램은 변수를 포함하는 조회를 실행할 수 있습니다. 다음 세 가지 방법 중 하나로 ISPF 서비스를 사용하는 응용프로그램에서 이러한 조회를 실행할 수 있습니다.

- ISPF 파일 조정 서비스를 사용합니다.

이 기술로 ISPF 파일 조정 구조에 따라 조회를 나타냅니다. 이러한 구조에서 변경할 수 있는 조회의 부분은 ISPF 대화 변수로서 나타납니다. 프로그램은 이 변수에 적절한 값을 부여한 후 특정 ISPF 파일 조정 서비스를 시작합니다. 그 결과 조회를 포함하는 순차적 파일이 나타납니다.

그런 다음, 프로그램은 조회를 QMF 임시 기억영역으로 가져와서 QMF가 실행하게 할 수 있습니다. 필수 IMPORT 및 RUN 명령은 호출 인터페이스 또는 명령 인터페이스를 통해 실행될 수 있습니다.

이 기술을 사용하려면, ISPF VDEFINE 서비스를 사용하여 프로그램에서 ISPF 대화 변수를 정의하는 방법을 알아야 합니다. *ISPF: Dialog Management Guide and Reference*를 참조하십시오.

- 프로그램 개발 기능(PDF) 편집기를 사용하여 QMF 오브젝트를 작성합니다.
조회, 프로시저, 서식 및 프로파일로 데이터 항목을 설계하고 제어하는 데 PDF 편집 매크로와 함께 PDF 편집기를 사용할 수 있습니다. REXX 프로그램을 사용하여 PDF 매크로를 작성할 수 있습니다.
- ISPF 대화를 사용하여 조회를 작성합니다.
SQL 조회를 포함하는 파일을 작성하기 위해 프로그램은 ISPF 표시장치 서비스를 사용하여 화면을 표시하고 사용자의 입력 항목에 따라 파일을 작성할 수 있습니다. 나중에 이 파일은 QMF로 가져와서 실행될 수 있습니다.

ISPF에서 QMF 논리를 갖는 프로시저로부터 프로그램 호출

ISPF에서 QMF를 실행할 경우 ISPF SELECT 서비스를 사용하여 논리를 갖는 프로시저로부터 호출 인터페이스 프로그램이나 REXX 프로그램을 호출해야 합니다. ISPF 대화 기능으로서 호출 인터페이스 프로그램을 실행하고 있음을 ISPF에 알리기 위해서는 PGM 키워드를 사용하십시오. 이 명령의 구문은 다음과 같습니다.

```
ADDRESS ISPEXEC "SELECT PGM(programname)"
```

REXX 프로그램의 경우, ISPF 대화 기능으로서 프로그램을 실행하고 있음을 ISPF에 알리기 위해서는 CMD 키워드를 사용하십시오. 이 명령의 구문은 다음과 같습니다.

```
ADDRESS ISPEXEC "SELECT CMD(cmdname)"
```

또는

```
ADDRESS ISPEXEC "SELECT CMD(cmdname parameters)"
```

cmdname은 호출 인터페이스 또는 REXX 프로그램의 이름입니다.

논리를 갖는 프로시저로부터 ISPF 명령 사용

ISPF에서 QMF를 시작할 때마다 QMF는 ISPF 프로그램으로서 시작됩니다. 따라서 ISPF에서 QMF 논리를 갖는 프로시저로부터 ISPF 명령을 실행하려면 QMF 프로그램 대화에서 ISPF 명령 대화로 이 명령을 전송해야 합니다. 이를 수행하려면 QMF 프로시저로부터 ISPF SELECT CMD를 발행해야 합니다.

올바른 ISPF 환경을 설정하고 ISPF 명령을 포함하는 REXX 프로그램을 실행하려면, CMD 키워드와 함께 다음 ISPF SELECT 명령을 사용하십시오.

```
ADDRESS ISPEXEC "SELECT CMD(userprogram)"
```

userprogram은 ISPF 명령을 포함하는 REXX 프로그램입니다.

예를 들어, ISPF 명령이 들어 있는 REXX 프로그램을 DIALOG라고 하면, 논리를 갖는 프로시저에 다음 명령을 포함시키십시오.

```
ADDRESS ISPEXEC "SELECT CMD(DIALOG)"
```

ISPF에 대한 자세한 내용은 *ISPF: Dialog Management Guide and Reference* 를 참조하십시오.

또한 QMF CMS 또는 TSO 명령을 사용하여 **CMS DIALOG** 또는 **TSO DIALOG**와 같은 ISPF 명령이 들어 있는 REXX 프로그램을 실행할 수도 있습니다. QMF는 사용자를 위해 ISPF SELECT CMD문을 발행합니다.

ISPF에서 QMF를 실행중이고 논리를 갖는 프로시저가 ISPF 서비스를 필요로 하는 프로그램을 시작하면 프로시저에서는 이전 예제에서 설명한 대로 ISPF SELECT CMD 환경을 사용하여 이들 프로그램을 시작해야 합니다. 예를 들어, ISPF에서 QMF를 실행하고 있으며 논리를 갖는 프로시저가 DB2의 DSN 명령을 시작한다고 가정해 보십시오. DSN 명령이 ISPF 서비스를 사용하므로, DSN 명령을 발행하려면 다음 명령 중 하나를 사용해야 합니다.

```
ADDRESS ISPEXEC "SELECT CMD(DSN)"
```

또는

```
ADDRESS ISPEXEC "SELECT CMD(DSNEXEC)"
```

여기서, DSNEXEC는 ADDRESS TSO DSN문을 포함합니다.

호출 인터페이스 고려사항

QMF 버전 7 이전에 링크 편집되고 호출 인터페이스를 사용하는 QMF 응용프로그램에서 LIBDEF 함수를 사용하려면, QMF 버전 7 인터페이스 모듈을 사용하여 응용프로그램을 다시 링크 편집해야 합니다.

ISPF에서 EDIT 명령 사용

ISPF에서 QMF 응용프로그램을 실행할 경우, 다음 명령을 사용하여 QMF SQL 조회 또는 프로시저를 편집할 수 있습니다.

```
EDIT QUERY
EDIT PROC
```

PROC 또는 QUERY 패널 내에서 QMF EDIT 명령을 발행하면, PROC 또는 QUERY 오브젝트 유형을 지정하지 않아도 됩니다. EDIT는 각각의 패널에서 호출할 때 이러한 값을 가정합니다. 기본적으로, QMF EDIT 명령은 PDF 편집기 세션에 프로시저 또는 조회를 추가합니다. QMF는 QMF 응용프로그램 ID *DSQn* 을 사용하여 PDF 편집기를 시작하는데 여기서 *n*은 NLF ID입니다. QMF는 QMF 응용프로그램과 일치되도록 기능 키와 명령행의 위치도 설정합니다.

이 기본값을 대체하려면, 다음과 같이 EDIT QUERY 및 EDIT PROC 명령을 사용하십시오.

```
EDIT QUERY (E=name)
EDIT PROC (E=name)
```

*name*은 다음 중 하나일 수 있습니다.

- 사용가능한 편집기
- DSQE가 아닌 응용프로그램 ID를 지정하는 REXX 프로그램의 이름. QMF가 제공한 것과 다른 기능 키를 설정하려는 경우 QMF 응용프로그램 ID와 다른 응용프로그램 ID를 사용할 수도 있습니다.

PDF PROFILE 데이터 세트 멤버를 필요로 하는 PDF EDIT 옵션을 사용할 경우 해당 멤버들을 작성해야 합니다. 예를 들어, PDF EDIT RECOVERY 옵션은 EDIT 명령을 사용하기 전에 존재해야 하는 *DSQnEDRT* PROFILE 데이터 세트 멤버를 필요로 합니다(여기서 *n*은 NLF 문자입니다).

ISPF를 사용하는 QMF 응용프로그램 작성

QMF EDIT 명령에 관한 자세한 내용은 온라인 도움말 및 *QMF* 참조서를 참조하십시오.

ISPF를 사용한 응용프로그램 디버깅

QMF 추적 기능은 QMF 메시지와 명령만 추적합니다. 응용프로그램의 ISPF 명령을 추적하려면 ISPF 로그 파일이나 데이터 세트에 메시지를 기록하십시오. 이 ISPF 서비스는 141 페이지의 『제9장 QMF 응용프로그램 디버깅』에서 설명된 QMF 추적 기능을 보완합니다.

ISPF 로그 서비스 사용

메시지를 ISPF 로그 파일에 작성하려면 ISPF 로그 서비스를 사용하십시오. 예를 들면, REXX에서 ISPF 로그에 메시지를 기록하는 ISPF 명령은 다음과 같습니다.

```
ADDRESS ISPEXEC LOG MSG (message-id)
```

*message-id*는 메시지 라이브러리에서 검색될 수 있고 로그에 작성될 수 있는 메시지 ID입니다.

PDF 대화 테스트 사용

PDF가 설치되어 있으면 대화 테스트 서비스(로그 옵션)를 사용하여 로그 파일이나 데이터 세트의 내용을 찾아볼 수 있습니다. ISPF를 빠져나가 로그 파일이나 데이터 세트를 인쇄할 수도 있습니다.

대화 테스트 서비스에는 응용프로그램 디버깅에 사용할 수 있는 많은 다른 유용한 옵션이 있습니다. 대화식으로 디버깅을 수행할 수 있습니다. 응용프로그램의 전체 또는 일부를 실행할 수 있고, 결과를 조사할 수 있고, 변경할 수 있고, 다시 실행할 수 있습니다. 대화 테스트를 사용하여 다음을 수행할 수도 있습니다.

- 선택 패널, 명령 프로시저 및 프로그램 시작
- 패널 표시
- 변수 추가 및 변수 값 수정
- ISPF 대화 서비스 실행
- 구분점 정의를 추가, 변경, 삭제
- 기능 및 변수 추적 정의를 추가, 수정, 삭제

대화 테스트 서비스의 추적(TRACES) 옵션을 사용하여 추적 정의를 작성, 변경 및 삭제할 수 있습니다. 따라서 대화 서비스 호출과 대화 변수 사용을 모니터링할 수 있습니다. 처리중에 충족되는 추적 정의가 나타나면 추적 출력이 ISPF 로그에 작성됩니다. 대화 테스트의 LOG 옵션을 사용하여 ISPF 로그를 찾아보거나 ISPF 종료시 인쇄된 출력을 검사할 수 있습니다.

특히 일반적인 ISPF 서비스와 대화 테스트에 대한 자세한 내용은 *ISPF Dialog Management Guide and Reference*를 참조하십시오.

제6장 2개 언어 응용프로그램 작성

많은 업무가 여러 다른 나라 또는 여러 나라 말을 사용하는 국가에서 처리되며, 이 경우 대화식 응용프로그램은 여러 개의 다른 자국어로 실행되어야 합니다. 버전 3.2를 사용하면 하나의 영어 응용프로그램을 작성하여 이를 QMF에서 지원하는 모든 자국어로 실행할 수 있습니다.

영어 이외에 언어에서 QMF 환경은 자국어 피처(NLF)를 수행합니다. NLF는 사용자에게 특정 언어로 조정된 QMF 세션을 제공합니다. 예를 들어, 독일어 NLF는 독일어 환경에서 QMF가 작동할 수 있게 해 줍니다.

QMF는 명령과 서식에 대해 2개 언어 지원을 제공합니다. 영어 QMF 명령을 실행하여 어떠한 NLF에서도 영어 서식을 표시하거나 번역가능한 응용프로그램을 작성할 수 있습니다. 이 장에서는 다국어나 비영어 환경에서 QMF로 작업하는 데 대한 정보를 제공합니다.

응용프로그램에 대한 2개 언어 오브젝트 작성

2개 언어 응용프로그램에서 오브젝트는 다른 QMF 오브젝트와 동일합니다. 즉, 영어로 오브젝트를 작성하고 저장하게 됩니다. 이러한 방법은 특정 오브젝트에 따라 달라질 수 있습니다.

조회 자국어로 프롬프트 표시나 QBE 조회를 작성할 수도 있고 영어로 SQL 조회를 작성할 수 있습니다.

서식 SAVE 명령에 있는 기본 언어(ENGLISH)나 기본적으로 설정된 언어를 사용하여 항상 기본 언어로 서식을 작성한 다음 저장합니다.

글로벌 변수 DSQEC_FORM_LANG는 SAVE 명령에 사용될 언어를 제어합니다. 기본값은 영어의 경우 1입니다. 0 값은 서식이 기본 세션 언어로 저장되도록 지정합니다.

프로시저

영어 또는 기본 언어로 프로시저를 작성할 수 있습니다.

2개 언어 응용프로그램 작성

SAVE 명령을 발행하여 NLF에서 작성하고 저장하는 서식은 영어로 변환할 수 있습니다. 예를 들면, 프랑스로 SEMAINE_F라는 서식을 영어로 WEEKLY_F인 서식으로 저장하는 명령은 다음과 같습니다.

```
SAUVER FORMAT SEMAINE_F EN WEEKLY_F (LANGUE=ANGLAIS
```

그 결과 NLF 서식을 2개 언어 응용프로그램에서 사용할 수 있는 영어 서식으로 전환됩니다.

명령 언어 변수 사용

응용프로그램에 필요한 오브젝트가 있으면 NLF 세션에서 영어 명령을 사용하여 시작할 수 있습니다. 이렇게 하려면, 기본 언어 변수 DSQEC_NLFCMD_LANG를 영어로 설정하십시오. 이 변수는 영어와 NLF 세션의 기본 언어간을 전환할 수 있게 합니다.

응용프로그램이 프로시저 이름 WEEKLY_P라고 가정하면, 다음 명령을 사용하게 됩니다.

```
"GET GLOBAL (CURR_LANG=DSQEC_NLFCMD_LANG"  
"SET GLOBAL (DSQEC_NLFCMD_LANG='1'"  
"RUN PROC WEEKLY_P"  
"SET GLOBAL (DSQEC_NLFCMD_LANG=CURR_LANG"
```

이 명령은 초기 프로시저에서 고급 언어 프로그램까지 유효한 QMF 응용프로그램의 일부가 될 수는 있지만 반드시 이 순서로 되어 있어야 합니다. 명령은 다음과 같은 방법으로 작동됩니다.

기본 언어 값 저장

GET GLOBAL 명령은 CURR_LANG라는 변수에 기본 언어 값을 저장합니다. 해당 값이 저장되면, DSQEC_NLFCMD_LANG를 영어 값 1로 재설정할 수 있습니다.

응용프로그램 실행

QMF 세션이 영어로 설정되어 있을 경우 영문 응용프로그램을 실행할 수 있습니다. 사용자가 입력하는 명령은 모두 영어여야 합니다. 그러나 사용자가 기능 키를 누르면 기본 명령은 기본 언어로 되어 있다고 가정됩니다.

QMF는 프롬프트 패널이 사용자의 기본 언어로 되어 있다고 가정합니다. EXPORT 및 IMPORT 명령 프롬프트 패널의 경우 기본 파일 유형 역시 기본 언어로 되어 있습니다.

NLF가 프로파일에서 대문자 옵션을 제공하면 QMF는 사용자가 영어 명령을 실행해도 사용자의 기본 언어 옵션을 고수합니다.

기본 언어로 리턴

응용프로그램을 종료한 후에 명령 언어 변수를 원래 값으로 다시 설정해야 합니다.

2개 언어 응용프로그램에서 초기 프로시저 사용

응용프로그램이 QMF를 시작하고 초기 프로시저를 실행하면 QMF는 사용자가 END 명령을 발행할 때마다 해당 프로시저를 실행합니다. 이 프로시저에 오류가 발생하면 QMF는 종료합니다. 예를 들어, 사용자가 영어로 실행하면서 기본 언어로 END 명령을 발행하면 QMF는 명령을 오류로 해석하고 종료됩니다.

다음 두 방법 중 하나로 이러한 상황을 피할 수 있습니다.

- 2개 언어 응용프로그램을 처리하도록 초기 프로시저를 변경합니다.

2개 언어 초기 프로시저는 그림9에 표시된 명령을 포함합니다.

```
"GET GLOBAL (CURR_LANG=DSQEC_NLFCMD_LANG"
"SET GLOBAL (DSQEC_NLFCMD_LANG=0"
:
:
/* QMF commands in the presiding language */
:
:
"SET GLOBAL (DSQEC_NLFCMD_LANG=CURR_LANG"
```

그림 9. 2개 언어 응용프로그램에서 초기 프로시저

- END 명령 후에 초기 프로시저를 실행하지 않습니다.

QMF에서 사용자가 END 명령을 발행할 때 초기 프로시저를 실행하지 않도록 변수 DSQEC_RERUN_IPROC를 0으로 설정할 수 있습니다.

영어 명령 사용

대부분 QMF 명령의 경우 영어로 명령을 실행하려면 먼저 기본 언어 변수를 변경해야 합니다. 기본 언어 변수가 영어로 설정되지 않은 경우에도 프롬프트 패널이 나 메시지를 표시하려면 NLF에서 일부 영어 명령을 실행해야 합니다.

예를 들어, 영어로 작성하고 NLF에서 실행되게 하려는 대화식 응용프로그램이 있는 경우 사용자에게 사용자 정의된 메시지를 제공하기 위해 MESSAGE 명령을 사용해야 합니다. 이 외에도, 다음 예제와 같이 프랑스어 NLF 세션에서 실행될 수 있는 메시지를 표시하려면, INTERACT 명령이 필요합니다.

```
proceed_text = 'Continue...'  
"RUN WEEKLY_Q"                /* Use the English RUN command */  
"SET GLOBAL (DSQEC_NLF_CMD_LANG=0" /* switch back to French */  
"MESSAGE (TEXT='proceed_text'" /* message in French */  
"INTERACT"                    /* show the report with message */
```

다음 명령은 모든 NLG에서 작동됩니다.

```
GET GLOBAL  
INTERACT  
MESSAGE  
SET GLOBAL  
START
```

다국어 환경

하나 이상의 NLF가 QMF에 설치되어 있으면 *다국어 환경*이 작성됩니다. 이러한 환경에서 적절한 권한 부여를 통해 각 QMF 세션에 대해 하나의 기본 언어를 선택할 수 있습니다. 예를 들어, 하나의 세션에 대해 영어를 선택할 수 있고 독일어 NLF가 설치되었으면 다른 세션에 대해 독일어를 선택할 수 있습니다. QMF 세션 중에 언어를 바꿀 수는 없으나 명령 언어 변수를 바꿀 수는 있습니다. 그런 다음에는 현재 세션을 종료해야 하며 적절한 언어 환경을 위해 다른 세션을 시작해야 합니다.

QMF 세션 환경

설치된 NLF가 없으면 QMF 세션 환경에서 사용할 수 있는 유일한 환경은 영어 언어 환경입니다. NLF가 설치되면 몇 가지 방식에서 NLF 환경은 영어 언어 환경과 다릅니다.

환경적 유사점

여러 면에서 QMF 세션 환경은 작동하는 NLF 유형에 관계없이 동일합니다. 가장 중요한 유사점은 다음과 같습니다.

가능

일반적으로, 영어 세션에서 수행할 수 있는 작업을 NLF 세션에서도 수행할 수 있습니다. 모든 임시 기억영역 오브젝트를 작성하고 저장할 수 있고, 보고서 형식을 지정하고 인쇄할 수 있고, SQL 명령을 발행할 수 있습니다. 프롬프트 조회, SQL 및 QBE 조회 및 QMF 프로시저도 실행할 수 있습니다. 영어와 NLF 환경 사이의 차이점은 가능한 작업 유형에 관계없이 실행하기 위해 터미널에 입력해야 하는 내용과 터미널 화면에 나타나는 언어에 있습니다.

SQL 및 QBE

SQL 및 QBE 언어의 명령어, 연산자 및 키워드는 변환되지 않습니다.

서식에 대한 사용 코드

이것은 동일합니다. 변환되지 않습니다.

시스템 명령

CMS, TSO 또는 CICS 및 ISPF 명령은 여전히 QMF에서 CMS, TSO 또는 CICS 명령을 통해 발행될 수 있습니다. 이 명령은 변환에 영향을 받지 않습니다. 즉, CMS, TSO 또는 CICS를 입력한 다음 실행할 명령을 입력하고 이를 QMF 외부에서 실행하는 경우와 똑같이 명령을 작성할 수 있습니다.

환경적 차이점

NLF 환경과 영어 언어 환경 사이의 보다 중요한 몇 가지 차이는 다음과 같습니다.

QMF 언어

모든 NLF에는 QMF 언어에 대한 완전한 명령어 및 키워드의 세트가 있습니다. 이들 명령어 및 키워드는 NLF 언어 환경에서 작동할 때 QMF 명령에 나타나야 합니다. 제공된 NLF의 경우, 이 단어는 번역될 수도 있습니다.

예를 들어, 독일어 NLF에서 명령어 DISPLAY와 키워드 PROC는 각각 ANZEIGEN와 PROZEDUR로 번역된다고 가정합니다. 독일어 세션중에 QMF는 명령 ANZEIGEN PROZEDUR을 이해하지만 DISPLAY PROC는 이해하지 못합니다.

QMF 언어의 어떤 요소는 명령 동의어이며 번역될 수 있습니다. 결과적으로, 각 NLF에는 고유하게 명명된 명령 동의어 표가 있습니다. LF가 설치되면 명령 동의어 표가 작성되고 NLF의 프로파일은 해당 NLF의 명령 동의어 표 이름을 표시합니다.

QMF 패널 및 메시지

모든 NLF에는 완전한 QMF 메시지 및 패널의 세트가 있습니다. QMF 명령의 명령어 및 키워드와 마찬가지로 이들은 번역되지 않을 수 있으나, 대부분의 경우 번역됩니다. 패널과 메시지에서 텍스트의 고정된 부분도 번역될 수 있습니다. 조회 이름과 같은 변수 정보는 번역되지 않습니다.

허용가능한 패널 입력

사용자 입력을 요구하는 프롬프트 패널과 서식 패널처럼 많은 QMF 패널은 어떤 항목의 범위를 작은 키워드 세트로 제한합니다. 허용되는 대부분의 값은 번역됩니다. 예를 들어, 영어 YES와 NO는 독일어 JA와 NEIN과 일치합니다.

프로파일 매개변수 값

다국어 환경에서, 사용자는 QMF 세션에 대해 사용할 수 있는 각 NLF마다 별도의 프로파일을 가집니다. 이 프로파일 각각에 대해 매개변수는 같고 같은 의미를 가집니다. 그러나 매개변수의 이름은 QMF가 제공하는 키워드 부분으로 번역됩니다. 어떤 매개변수의 경우, 매개변수가 가정할 수 있는 값도 역시 번역됩니다.

예를 들어, 영어 프로파일에서 CASE 매개변수는 값 UPPER, STRING 또는 MIXED를 가질 수 있습니다. 독일어 프로파일에서, CASE 매개변수는 SCHRIFT 매개변수이고 가정하는 값은 GROSS, KETTE 및 GEMISCHT입니다.

내보내어 저장된 서식 오브젝트

SAVE, EXPORT 및 IMPORT 명령은 저장된 서식 오브젝트를 원하는 언어로 지정하게 합니다. 그것을 영어로 저장하거나 현재 세션의 기본 언어로 저장할 수 있습니다. 이 명령에 관한 자세한 내용은 *QMF* 참조서를 참조하십시오.

샘플표 및 조회

IBM에서 몇 가지 NLF와 함께 영어 샘플표 및 조회의 번역된 버전을 제공하는 경우가 있습니다. 예를 들어, 일본어 사용자는 영어 표에서 번역된 샘플 표를 가지게 됩니다.

번역할 수 있는 응용프로그램 작성

가능한 많은 언어 감지 오브젝트에 대한 변수를 사용하여 응용프로그램을 새 언어에 맞게 조정하는 시간을 크게 절약할 수 있습니다. 이러한 변수는 다음을 포함할 수 있습니다.

- QMF 명령에서 명령어, 오브젝트 이름 및 옵션 ID
- 설치시 정의된 패널 이름

응용프로그램에 대해 자체의 패널을 작성할 경우 응용프로그램이 실행될 때 사용되는 각 언어에 대해 번역된 패널의 세트가 필요합니다. 이 패널에 고유 이름을 설정하고 응용프로그램 사용자가 사용할 수 있게 하십시오. 그러면 응용프로그램은 패널 이름에 대한 변수를 사용할 수 있습니다.

- 설치시 정의된 메시지 ID

2개 언어 응용프로그램 작성

패널처럼 메시지도 적절한 NLF 언어로 번역되어야 합니다. 응용프로그램은 메시지 이름에 대해 변수를 사용할 수 있습니다.

변수를 사용하면 몇 가지 NLF에서 같은 프로그램을 사용할 수 있습니다.

제7장 응용프로그램의 QMF 명령

특정 환경에서 QMF 명령행에 유효한 모든 명령은 응용프로그램에도 유효합니다. 또한 QMF는 응용프로그램용으로 특별히 설계된 명령을 제공합니다.

이 장에서는 사용자가 일반적으로 프로그램에 포함시키는 QMF 명령을 설명하고 응용프로그램 개발시의 명령 사용법에 대해 설명합니다. 명령 및 명령 구문에 관한 자세한 내용은 *QMF 참조서*를 참조하십시오.

CONNECT

QMF CONNECT 명령을 사용하여 QMF 세션 중에 원격 작업 단위로 분산된 네트워크 내의 다른 시스템에 연결할 수 있습니다. QMF CONNECT 명령을 사용하여 QMF에서 지원하는 원격 데이터베이스에 액세스할 수도 있습니다. 원격 시스템에 연결하면, 이 시스템이 **현** 위치가 됩니다. 응용프로그램을 작성할 때, 다음에서 이 명령을 발행할 수 있습니다.

- 호출 인터페이스
- 명령 인터페이스
- 프로시저 내부(선형 또는 논리)

QMF CONNECT 명령을 사용하여 원격 작업 단위를 시작할 경우 응용프로그램의 특정 부분에 영향을 줄 수 있습니다. 다음 고려사항에 유의하십시오.

- 응용프로그램이 새 위치로 연결될 경우 QMF 프로파일, 명령 동의어, 기능 키는 새(현) 위치 값으로 다시 초기화됩니다.
- QMF를 시작하고 QMF 명령을 발행하는 모든 호출 인터페이스와 명령 인터페이스 프로그램은 사용자(지역 시스템)와 같은 시스템에 상주합니다. 프로그램은 지역 시스템에서 QMF를 시작한 후에 원격 데이터베이스에 연결하기 위해 QMF CONNECT 명령을 발행할 수 있습니다. 데이터베이스 오브젝트에 영향을 주는 모든 후속 QMF 명령 및 SQL 문은 **현** 위치(원격 데이터베이스)에서 실행됩니다.

- QMF에 의해 시작되는 모든 프로그램은 QMF가 실행중인 운영 체제(지역 시스템)의 규칙을 따라야 합니다.
- 다른 유형의 명령들은 원격 작업 단위와 상당히 다른 방식으로 작동합니다. 응용프로그램이 원격 작업 단위를 사용할 경우 모든 시스템 고유 명령과 대부분의 QMF 명령은 QMF가 실행되고 있는 시스템(일반적으로, 지역 시스템)에서 실행됩니다. 그러나 QMF 명령이 다음 중 하나를 수행할 경우:
 - SQL 명령을 데이터베이스에 보냅니다.
 - 데이터베이스에 저장된 QMF 오브젝트와 데이터를 사용하거나 변경합니다.이 명령은 현 위치에 있는 데이터베이스에 영향을 줍니다.

예제

사용자는 CMS를 실행하는 지역 VM 시스템(SANJOSE)으로 로그인됩니다. 다음을 수행하는 REXX 호출 인터페이스 프로그램을 작성하고자 합니다.

1. QMF 세션을 시작합니다.

```
CALL DSQCIX "START"
```

2. 원격 DB2 데이터베이스(DALLAS)에 연결합니다.

```
CALL DSQCIX "CONNECT TO DALLAS"
```

3. 원격 데이터베이스에서 데이터를 조회하고, 데이터를 형식화하고, 보고서를 인쇄하는 논리를 갖는 프로시저(EARNINGS)를 실행합니다.

```
CALL DSQCIX "RUN PROC EARNINGS"
```

EARNINGS 프로시저에는 다음과 같은 논리가 포함되어 있습니다.

```
:  
:  
"RUN QUERY EARNQ (FORM=EARNF"  
"PRINT REPORT"  
:  
:
```

이 프로시저는 다른 CONNECT 명령을 포함하지 않습니다.

4. QMF 세션을 종료합니다.

```
CALL DSQCIX "EXIT"
```

이 프로그램을 작성할 경우, 다음 사항에 유의하십시오.

- 응용프로그램은 지역(SANJOSE) VM 시스템에 상주합니다.

- QMF 세션은 지역(SANJOSE) VM 시스템에서 시작됩니다.
- 프로시저는 원격 데이터베이스(DALLAS)에 상주해야 합니다. DALLAS는 60 페이지의 3단계에서 응용프로그램이 프로시저를 실행할 경우 현 위치입니다.
- 60 페이지의 2단계에서 CONNECT 명령 다음에 응용프로그램이나 프로시저에서 사용되는 모든 QMF 오브젝트(이 경우, 조회 및 서식)는 원격 데이터베이스(DALLAS)에 상주해야 합니다.
- 60 페이지의 3단계에서 프로시저가 실행하는 SQL 조회 EARNQ는 DALLAS의 DB2 데이터베이스에 대해 실행됩니다.
- 프로시저 EARNINGS에서 PRINT 명령은 현 위치(DALLAS)에 프로파일에 의해 명명된 프린터에서 보고서를 인쇄합니다. 이 예제의 경우, 현 위치(DALLAS)에 있는 프로파일은 프린터가 지역 VM 시스템(SANJOSE)에 있는 것으로 정의한다고 가정하십시오.

QMF CONNECT 명령을 사용한 원 위치로의 연결에 대한 자세한 내용은 온라인 도움말을 참조하십시오.

END

QMF 세션을 종료하려면 응용프로그램에 END 명령을 포함할 수 있습니다. 일반 사용자가 종료 키를 누르거나 명령행에 END 명령을 입력하여 대화식 QMF 세션을 종료한 후 응용프로그램 작업 상태로 리턴하도록 응용프로그램을 설계할 수도 있습니다.

END 명령을 제어하는 규칙은 END 명령이 발행되는 세션 유형에 따라 달라집니다. 이 절에서는 다음 QMF 세션 유형 각각에서 END 명령이 작동하는 방법에 대해 논의합니다.

- 호출 인터페이스에 의해 시작되는 세션
- 초기 프로시저로 ISPF를 사용하는 대화식 세션
- 초기 프로시저 없이 ISPF를 사용하는 대화식 세션
- INTERACT 명령에 의해 시작되는 대화식 세션
- 일괄처리 모드 세션

호출 인터페이스에 의해 세션 시작

호출 인터페이스에 의해 시작되는 대화식 세션에서 일반 사용자가 END 명령을 발행하면 대화식 세션이 종료되고 제어는 호출하는 응용프로그램 작업 상태로 리턴됩니다. QMF는 사용중인 세션을 종료하기 전에 시작 패널을 현재 패널로 만듭니다. QMF는 사용중입니다. QMF가 호출 인터페이스 응용프로그램에 의해 시작된 후에는 EXIT 명령 또는 심각한 오류가 발생했을 때에만 QMF가 종료됩니다.

초기 프로시저(DSQSRUN)가 있는 대화식 세션

QMF에서는 QMF가 이러한 키워드를 사용하여 시작될 때 초기 프로시저를 실행하는 대화식 세션을 시작합니다.

```
DSQSRUN=xxxxx,DSQSMODE=I
```

여기서, xxxxx(DSQSRUN 키워드의 값)는 QMF 초기 프로시저의 이름입니다. 이 키워드에 대해서는 78 페이지의 『START 명령 키워드』에서 설명됩니다.

QMF는 시작한 후에 초기 프로시저를 실행합니다. 이 프로시저가 종료된 후에 현재 패널이 시작 패널이 아니면 사용자는 대화식 세션에 있습니다. 현재 패널이 시작 패널이면 QMF는 대화식 세션을 시작하지 않습니다. 대신, QMF에서는 다음 명령문이 모두 적용되면 바로 초기 프로시저를 다시 시작합니다.

- 심각한 오류가 발생하지 않았습니다.
- DSQEC_RERUN_IPROC 글로벌 변수가 1로 설정됩니다.

프로시저의 끝에서 현재 패널이 시작 패널이 되려면 초기 프로시저를 작성하지 않아야 합니다. 시작 패널이 초기 프로시저의 끝에서 현재 패널이면 연속 루프가 발생합니다. QMF는 초기 프로시저를 시작하거나 실행하지 않는 것처럼 나타낼 수 있습니다. 이렇게 하지 않으려면, 다음 중 하나가 발생하는지 확인하십시오.

- 프로시저의 끝에 있는 현재 패널이 시작 패널이 아닙니다.
- 프로시저는 QMF EXIT 또는 INTERACT 명령 중 하나를 포함합니다.

일반 사용자가 대화식 세션에서 END 명령을 발행하고 DSQEC_RERUN_IPROC=1이면, QMF에서는 간단히 초기 프로시저를 다시 시작합니다. 세션을 종료하려면 EXIT 명령을 사용하십시오.

QMF가 호출 인터페이스를 통해 시작되지 않으면, DSQ_RERUN_IPROC를 사용하여 QMF의 초기 프로시저 재실행 여부를 제어할 수 있습니다.

DSQEC_RERUN_IPROC = 0을 설명하면, QMF에서는 END 명령 실행시 초기 프로시저를 다시 실행하지 않고 종료합니다. 이 변수는 호출 인터페이스 응용프로그램에 어떠한 영향도 미치지 않습니다.

초기 프로시저가 없는 대화식 세션

이 경우, DSQSRUN 매개변수는 QMF가 시작될 때 지정되지 않습니다. 이것은 사용자가 제어를 얻기 전에 어떠한 프로시저도 실행되지 않게 합니다.

일반 사용자가 이 대화식 세션 내에서 END 명령을 발행하면, QMF에서는 다음 중 하나를 수행합니다.

- 현재 패널이 다른 패널이면 시작 패널을 현재 패널로 만듭니다.
- 현재 패널이 시작 패널이면 세션 종료를 표시합니다. 온라인으로 실행되는 경우 END 명령은 즉시 세션을 종료합니다. 응용프로그램에서 실행되는 경우 응용프로그램이 종료될 때마다 세션이 종료됩니다.

INTERACT 명령에 의해 시작되는 대화식 세션

응용프로그램은 66 페이지의 『INTERACT』에 설명된 INTERACT 명령을 사용하여 현재의 대화식 QMF 세션에서 새로운 대화식 QMF 세션을 시작할 수 있습니다. 초기 프로시저의 유무에 관계없이 이전 세션은 1차 세션이거나 다른 응용프로그램에 의해 시작되는 세션이 될 수 있습니다.

새 세션에서, 사용자는 END 명령을 온라인으로 입력하거나 응용프로그램이 명령을 발행할 수 있습니다. 현재 패널 유형에 관계없이 END 명령 실행은 세션 종료를 표시합니다.

온라인으로 발행되는 경우, END 명령은 즉시 세션을 종료합니다. 응용프로그램에서 실행되는 경우 응용프로그램이 종료될 때마다 세션이 종료됩니다. 세션이 종료될 경우 세션을 시작했던 응용프로그램 작업 상태로 리턴됩니다.

일괄처리 모드 세션

QMF 일괄처리 모드 세션은 QMF에서 지원되는 모든 환경에서 비대화식 세션에서 실행됩니다. 기본 옵션인 일괄처리 모드(DSQSMODE=BATCH)를 지정하여 QMF를 시작하고 화면이 표시되지 않도록 할 수 있습니다. DSQQMFE를 사용할

응용프로그램의 QMF 명령

경우 DSQSRUN을 사용하여 초기 프로시저를 지정해야 합니다. 그러나 호출 인터페이스를 사용할 경우 초기 프로시저를 지정하지 않아도 됩니다.

일괄처리 QMF 세션중에 초기 프로시저는 END 명령을 발행하거나 END 명령을 발행하는 응용프로그램을 시작할 수 있습니다. 결과는 초기 프로시저가 없는 대화식 세션의 경우와 같습니다. END 명령은 다음을 수행합니다.

- 다른 패널이 현재 패널이면 시작 패널을 현재 패널로 만듭니다.
- 현재 패널이 시작 패널이면 세션 종료를 표시합니다.

초기 프로시저에서 발행되면 END 명령은 세션을 즉시 종료합니다. 응용프로그램에서 발행되면 응용프로그램이 QMF EXIT 명령을 발행할 때마다 세션이 종료됩니다.

세션중에 어떠한 대화도 허용되지 않습니다. 따라서 해당 세션이 새 세션을 시작할 수 없습니다.

EXIT

EXIT 명령은 QMF 세션 시작 방법에 관계없이 똑같이 작동하며, 모든 사용자의 세션 종료를 표시합니다. 일괄처리 모드에서는 단지 하나의 세션만 있습니다. 대화식 세션의 경우, 여기에는 1차 세션이 포함되고 모든 세션은 INTERACT 명령에 의해 시작됩니다.

EXIT 명령이 명령행에 입력되면 명령이 입력된 세션은 즉시 종료됩니다. INTERACT 명령에 의해 시작된 각 세션은 세션을 시작한 응용프로그램이 종료될 때 종료됩니다. EXIT 명령이 응용프로그램에서 발행되면 원래의 QMF 세션이 종료할 때 해당 세션이 종료됩니다. INTERACT 명령에 의해 시작되는 모든 대화식 세션은 QMF가 종료하기 전에 종료됩니다.

호출 인터페이스 프로그램에서 응용프로그램이 QMF를 사용하여 완료되었을 경우 QMF EXIT 문을 포함시키는 것이 중요합니다. 이 명령을 포함시키지 않으면 QMF 세션은 로그오프되거나 일괄처리 작업이 완료될 때까지 사용중인 상태를 유지합니다.

사용자나 응용프로그램에서 EXIT 명령을 발행하면, QMF에서는 DSQAO_TERMINATE를 1로 설정합니다(종료 표시). QMF에서 실행되는 응용프로그램만 테스트될 수 있고 이 글로벌 변수를 사용할 수 있습니다. QMF가 주 QMF 세션으로 리턴될 때 DSQAO_TERMINATE가 1로 설정되면, QMF는 즉시 종료되며 자원을 사용가능하게 만듭니다.

GET GLOBAL

GET GLOBAL 명령을 사용하여 응용프로그램에서 QMF 글로벌 변수를 액세스할 수 있습니다. REXX 이외의 언어의 경우, QMF는 GET GLOBAL 명령의 확장 구문을 제공합니다.

```
▶▶—GET Global—(—| Variable definitions |—————▶▶
```

Variable definitions:

```
|—number of varnames—,—varname lengths—,—varnames—,——————|
```

```
|—value lengths—,—values—,—value type——————|
```

GET GLOBAL 명령에 지정된 매개변수는 응용프로그램이 GET GLOBAL 명령에 의해 리턴되는 변수 이름 및 값을 저장하기 위해 사용하는 기억영역을 정의합니다.

number of varnames

요청되는 변수의 수

varname lengths

지정된 각 변수 이름에 대한 길이 목록

변수 이름의 길이는 기억영역의 글로벌 변수 이름의 실제 길이와 같습니다. 맨 뒤 공백으로 채워지는 18자의 영역이 허용됩니다.

varnames

QMF 변수의 이름 목록

QMF는 맨 뒤 공백을 삭제하기 때문에 글로벌 변수 이름에 맨 뒤 공백을 지정하면 안 됩니다.

value lengths

변수 값의 길이 목록

다음 규칙은 변수 값에 적용됩니다.

- 제공한 값 길이가 QMF에 저장된 값의 길이보다 짧으면 QMF는 오른쪽 값을 절단하고 잘린 값을 리턴합니다.
- 제공한 값 길이가 QMF에 저장된 값의 길이보다 길면 QMF는 맨 뒤 공백으로 채워진 값을 리턴합니다.
- 정수 길이는 항상 4바이트입니다.

values

변수 값 목록

value type

값을 포함하는 기억영역의 데이터 유형. 이 값은 문자나 정수 중 하나여야 합니다.

INTERACT

INTERACT 명령은 일반 사용자를 대화식 QMF 또는 GDDM ICU 세션에 들어가게 합니다. 이 세션에 있는 동안 일반 사용자는 이 제품의 정상적인 대화식 세션중에 있는 것처럼 명령을 입력할 수 있습니다.

INTERACT에는 세션 및 명령의 두 가지 서식이 있습니다.

INTERACT의 세션 서식

INTERACT 명령을 발행하면 QMF는 현재 패널에 사용자를 배치하고 사용자가 QMF 명령을 대화식으로 발행하게 합니다. INTERACT 명령은 현재 세션 내부에 다른 QMF 『세션』을 제공합니다. INTERACT 명령은 사용자를 대화식 QMF 세션이나 대화식 GDDM ICU 세션 중 하나에 배치할 수 있습니다.

- 대화식 QMF 세션의 경우

일반적으로 QMF 패널을 보여주는 QMF 명령 다음에 INTERACT 명령을 발행합니다. 이 세션에서 사용자는 대화식 QMF에서 유효한 어떠한 명령도 입력할 수 있습니다.

- 대화식 GDDM ICU 세션의 경우

일반적으로 QMF가 GDDM ICU를 시작하게 하고 ICU 패널을 보여주는 명령 다음에 INTERACT 명령을 발행합니다. 이 세션에서 사용자는 ICU에서 유효한 어떠한 명령도 입력할 수 있습니다.

시나리오

한 단계만으로 보고서를 작성해야 하는 프로시저를 실행할 경우, 다음과 같습니다.

```
/* This procedure prints the weekly sales report. */
"RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"
"PRINT REPORT"
```

그림 10. INTERACT 명령이 없는 단순 프로시저

QMF는 "프로시저가 실행되었습니다."라는 메시지와 함께 형식화된 데이터가 들어 있는 REPORT 패널을 표시합니다.

그러나 여러 단계로 된 프로시저를 작성할 수도 있습니다. 프로시저의 중간 결과를 보려면 INTERACT 명령을 사용하십시오. 둘 이상의 조회를 실행하는 프로시저의 중간 결과를 보려면, 첫번째 RUN 명령 바로 다음에 INTERACT 명령을 삽입하십시오.

```
/* This procedure generates a report showing annual sales. */
"RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"
"INTERACT"
"RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F"
```

그림 11. 프로시저에서 INTERACT 사용

그런 다음 시작 패널에서 이 프로시저를 실행할 때 QMF는 형식화된 데이터를 포함하는 REPORT 패널을 보여줍니다. 다음에 REPORT 패널에서 END 명령을 입력하면 프로시저는 계속되어 두 번째 조회를 실행하고 마지막 보고서를 표시합니다. INTERACT 명령을 생략하면 QMF는 첫번째 조회의 결과를 보여주지 않고 마지막 보고서만 나타냅니다.

INTERACT 명령은 REXX에서 동일한 명령이 다음과 같은 경우에도 호출 인터페이스를 통해 발행될 때 같은 결과를 얻습니다.

```
⋮  
call dsqcix "RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"  
call dsqcix "INTERACT"  
call dsqcix "RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F"  
⋮
```

그림 12. REXX 응용프로그램에서 INTERACT 사용

Call dsqcix "INTERACT" 행은 호출 인터페이스를 통해 INTERACT 명령을 발행하는 REXX 구문입니다. 호출 인터페이스를 통해 INTERACT 명령을 발행하려면 프로그래밍 언어가 필요로 하는 구문을 사용해야 합니다.

보고서 화면 출력 억제

QMF 호출 인터페이스 응용프로그램에서 조회를 실행할 경우 기본적으로 QMF는 결과 보고서를 표시합니다. 그러나 DSQDC_DISPLAY_RPT 글로벌 변수를 0으로 설정하여 결과 보고서를 자동으로 표시하지 않도록 QMF에 지시할 수 있습니다. DSQADPAN=0을 지정하여 START 명령에 이 글로벌 변수도 설정할 수 있습니다.

RUN QUERY 명령이 응용프로그램으로부터 발행될 경우에만 이 글로벌 변수를 사용할 수 있습니다. QMF 명령행으로부터 RUN QUERY가 발행된 경우 보고서의 화면 출력에 영향을 주지 않습니다.

INTERACT 세션 종료

사용자가 END 명령을 발행하면 INTERACT 명령을 발행한 프로세스 작업 상태로 리턴됩니다. 그러나 두 세션은 독립적이지 않습니다. 이전 세션이 다시 시작될 경우 INTERACT 세션중에 수행된 모든 결과는 그대로 보존됩니다. 예를 들어, 사용자가 새 대화식 세션에서 현재 서식 오브젝트를 변경할 경우 새 세션이 종료되면 이전 세션에 있는 현재 서식 오브젝트에는 이 변경사항이 적용됩니다.

사용자가 QMF 오브젝트 패널에서 END 명령을 발행한 후에 응용프로그램이 QMF 시작 패널을 표시하게 하려면(대화식 QMF가 표시하는 방식으로) 249 페이지의 『INTERACT 루프 사용에 대한 REXX 예제』의 논리를 추가하십시오.

INTERACT의 명령 서식

명령 인터페이스 응용프로그램이 INTERACT의 명령 서식을 사용하고 QMF가 대화식 세션(DSQSMODE=I)을 실행한 경우에만 명령 인터페이스(DSQCCI)는 QMF 명령을 대화식으로 실행합니다.

INTERACT의 명령 서식은 호출 인터페이스를 통해 발행되는 명령에 어떠한 영향도 미치지 않습니다. 호출 인터페이스에서 명령이 대화식으로 실행될지 여부를 제어하는 유일한 방법은 START 명령 키워드 DSQSMODE를 설정하는 것입니다. DSQSMODE 키워드에 관한 자세한 내용은 80 페이지의 표7을 참조하십시오.

지정된 명령의 대화식 실행을 요청하려면 다음 구문을 사용합니다. 다음 명령을 실행하십시오.

`INTERACT command`

여기서 *command*는 지정된 명령을 나타냅니다. QMF와 사용자간의 명령 실행에 대해 어떠한 대화가 발생하면 QMF는 이 명령을 대화식으로 실행합니다. 다양한 QMF 프롬프트 및 상태 패널이 이 대화에서 나타날 수 있습니다.

예를 들어, 다음 명령은 RUN QUERY 명령 옵션의 명령 프롬프트 패널을 표시합니다.

`INTERACT RUN QUERY ABC ?`

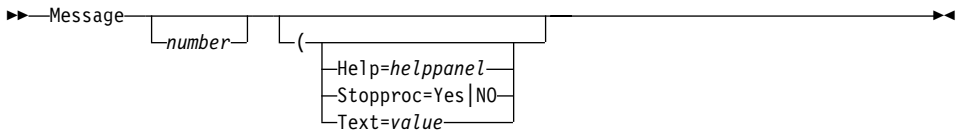
QMF 일괄처리 세션으로서 대화식 실행이 허용되지 않으면 INTERACT의 명령 서식은 선행 명령에 어떠한 영향도 미치지 않습니다.

DSQAO_INTERACT 변수를 검사하여 현재 세션에서 대화식 실행이 허용되는지를 검사할 수 있습니다. 1값은 INTERACT가 허용된다는 것을 의미합니다. 예를 들어, 일괄처리 응용프로그램은 대화식 실행을 허용하지 않습니다. 부록F. QMF 글로벌 변수 표의 287에서 DSQAO_INTERACT에 대한 자세한 내용을 참조하십시오.

MESSAGE

응용프로그램을 작성할 때 사용자에게 표시되는 정보나 다음에 수행해야 할 기능에 대한 특정 메시지를 사용자에게 제시하려는 경우가 종종 있습니다. MESSAGE 명령을 통해 QMF 패널에 자신의 메시지를 작성하고 메시지를 표시할 수 있습니다. ISPF에서 QMF가 ISPF 오류 메시지에 대해 메시지 도움말을 표시하도록 지정할 수도 있습니다.

MESSAGE 명령 구문은 다음과 같습니다.



number(ISPF에만 유효)

*number*는 ISPF에서만 유효합니다. 이 매개변수는 ISPF 메시지 라이브러리에 서 메시지 정의의 수를 나타냅니다.

HELP

이 상황에서 일반적으로 표시되는 메시지로 정의되지 않은 도움말 패널을 지정하려면 이 매개변수를 사용하십시오. *helppanel*을 적절한 패널 ID로 바꾸십시오.

QMF 패널을 표시하는 경우 패널 정의가 DSQPNLE에 있으므로 패널을 변경할 수 없습니다.

ISPF에서 자신의 패널을 작성하고 표시하려면 패널의 정의가 ISPF 패널 라이브러리에 있어야 하고 이 라이브러리는 ISPLIB 파일 또는 데이터 세트로 연결되어야 합니다. 패널은 메뉴 또는 데이터 항목 패널이 아니라 도움말 패널이어야 합니다.

ISPF에서 *number*를 지정했다면 *helppanel*의 기본값은 *number*에 의해 지정된 메시지 정의에 대한 도움말 패널 표시기가 됩니다.

ISPF에서, *number*에 의해 지정된 메시지 정의가 도움말 패널 표시기를 정의하지 않을 경우 MESSAGE 명령은 메시지 도움말을 제공하지 않습니다. 대신, 사용자가 도움말을 요청하면 오브젝트 패널에 대한 QMF 도움말이 화면에 나타납니다.

STOPPROC

프로시저 종료 스위치를 설정하여 선형 프로시저의 실행을 억제하려면 Stopproc를 사용하십시오. 다음 명령은 프로시저 종료 스위치를 설정합니다.

```
Message (Stopproc=Yes
```

Stopproc=Yes이면 프로시저 종료 스위치는 켜집니다. 기본값은 No(꺼짐)입니다. 이 스위치는 선형 프로시저에만 영향을 줍니다.

이 스위치가 켜져 있는 동안 제어를 받는 모든 QMF 프로시저는 실행이 종료됩니다. 이 스위치가 꺼져 있는 동안에는 프로시저가 정상적으로 실행됩니다.

스위치가 꺼져 있는 경우 MESSAGE 명령을 실행해야만 이 스위치를 켤 수 있습니다. 스위치가 켜져 있는 경우, 다음 중 하나가 발생할 때까지 그대로 켜져 있습니다.

- 다른 QMF 명령이 실행됩니다. 이 명령은 스위치를 켜려는 옵션을 가진 MESSAGE 명령을 제외하고 모든 QMF 명령이 될 수 있습니다.
- 응용프로그램이 종료된 경우 사용자가 제어할 수 있는 상태가 됩니다. 사용자는 항상 QMF 프로시저를 실행하는 온라인 명령을 실행할 수 있습니다.

DSQCM_MESSAGE 변수를 검사하여 프로시저 종료 스위치가 켜져 있는지 검사할 수 있습니다. 종료 옵션이 적용되면 이 변수는 종료 스위치를 켜는 MESSAGE 명령에 대한 메시지를 포함합니다.

TEXT 옵션

ISPF 메시지 정의에서 메시지를 정의하거나 텍스트를 덮어쓰려면 TEXT=를 사용하십시오. *value*를 메시지에 사용된 문자열로 바꾸십시오. 공백 문자가 들어 있는 값은 구분문자로 묶여 있어야 합니다. 메시지 값의 유효한 구분문자는 작은 따옴표, 괄호 및 큰 따옴표입니다. 구분문자가 큰 따옴표인 경우, 큰 따옴표는 메시지의 일부로 표시됩니다. 메시지 값의 최대 길이는 78개의 1바이트 문자입니다. 78자보다 긴 값은 처음 78자만 포함되도록 잘립니다. QMF는

| 텍스트를 대문자로 표시하지 않습니다. 그러나 DSQCCI(명령 인터페이스)를 통
 | 해 MESSAGE가 발행되면 ISPF는 텍스트를 대문자로 표시할 수도 있습니다.
 | 메시지가 따옴표로 묶이면 **TEXT=** 세부사항을 큰 따옴표로 묶어야 합니다.
 | ISPF에서 기본값은 *number*에 의해 지정된 ISPF 메시지의 긴 메시지 텍스트
 | 가 됩니다. 텍스트는 그대로 남아 있습니다. 사용자의 QMF 프로파일에 대해
 | 설정된 CASE의 값이 무엇이든지 어떠한 폴딩도 발생하지 않습니다.

MESSAGE 명령을 사용한 메시지 생성 예제

프로시저를 사용하여 두 조회를 실행하고 두 보고서를 표시하는 응용프로그램을 작
 성한다고 가정해 보십시오. QMF가 첫번째 보고서를 표시할 때 사용자가 두 번째
 보고서를 진행할 준비가 되어 있을 경우 대화식 세션을 종료하라는 메시지를 사용
 자에게 표시해야 할 것입니다. 그림13과 같은 선형 프로시저를 작성할 수 있습니
 다. 이것은 REPORT 패널에 있는 MESSAGE 명령에 의해 정의된 메시지를 포
 함합니다. REPORT 패널에 메시지를 표시하려면, INTERACT 명령 바로 앞에
 MESSAGE 명령을 두십시오.

```

:
RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F
MESSAGE (TEXT='OK, press END when you are finished viewing this report.'
INTERACT      RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F
:
    
```

그림 13. MESSAGE 명령 사용 예제

QMF에서 이 프로시저의 INTERACT 명령을 처리하면, REPORT 패널은 다음과 같이 표
 시됩니다.

전체 수입: -

Employee: JONES
Position: MGR

1=도움말	2=	3=종료	4=인쇄	5=차트	6= 조회
7=앞화면	8=뒤화면	9=서식	10=왼쪽	11=오른쪽	12=

이 보고서 보기를 종료하려면 종료를 누릅니다.
명령 ==>

논리를 갖는 프로시저를 사용할 경우 그림14와 같이 텍스트 문자열 대신 REXX 변수를 사용할 수 있습니다. REXX 변수를 사용할 경우 *messagetext* 텍스트 문자열에서 변수 이름을 큰 따옴표로 묶습니다.

```
oktext = 'OK, press END when you are finished viewing this report.'
"RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"
"MESSAGE (TEXT='oktext'"
"INTERACT"
"RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F"
```

그림 14. 프로시저에서 MESSAGE 명령과 함께 REXX 변수 사용

ISPF와 함께 사용할 수 있는 MESSAGE 명령 예제

MESSAGE MSG011X

- MSG011X에서 메시지 텍스트는 긴 메시지입니다.
- MSG011X에서 메시지 도움말 패널은 확인된 패널입니다(있는 경우).
- QMF가 명령을 처리한 후에 프로시저 종료 스위치가 설정될지 여부는 MSG011X에서 프로시저 종료 스위치에 의해 결정됩니다.

MESSAGE MSG011X (HELP=ANELX STOPPROC=YES

- MSG011X에서 메시지 텍스트는 긴 메시지입니다.
- 메시지 도움말 패널은 PANELX로 명명된 패널입니다.
- 프로시저 종료 옵션 스위치는 켜지고 응용프로그램에서 QMF 선행 프로시저의 실행을 억제합니다.

SET GLOBAL

SET GLOBAL 명령으로 자신의 글로벌 변수를 작성할 수 있고 QMF 명령에서 글로벌 변수를 대체 변수처럼 사용할 수 있습니다. 사용자는 자신의 고유 글로벌 변수를 사용할 수 있으며 QMF가 제공한 글로벌 변수를 사용할 수도 있습니다. QMF에 제공된 글로벌 변수의 목록에 대해서는 285 페이지의 『부록F. QMF 글로벌 변수 표』를 참조하십시오.

프로시저에 글로벌 변수를 설정하려면, 다음 중 하나를 수행하십시오.

- SHOW GLOBALS 패널에서 변수를 설정합니다.

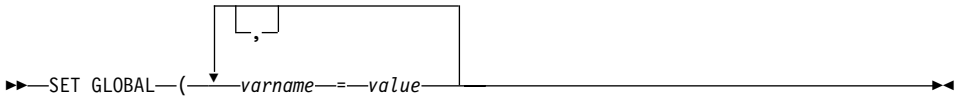
변수 이름은 최대 18자까지 될 수 있으며 변수의 길이는 최대 32 768자까지 될 수 있습니다.

응용프로그램의 QMF 명령

- 명령행 또는 SET GLOBAL 프롬프트 패널에서 프로시저에 SET GLOBAL 명령의 선형 구문을 사용합니다.
- REXX 이외의 호출 인터페이스 언어에 대해서는 확장 구문을 사용합니다. 호출 인터페이스에서 SET GLOBAL 명령의 확장 구문을 사용하는 데 대한 자세한 정보는 SET GLOBAL: 확장 구문을 참조하십시오.

SET GLOBAL: 선형 구문

글로벌 변수의 이름은 최대 17자까지 될 수 있고 값의 길이는 55자 이하입니다. SET GLOBAL 명령의 선형 구문은 다음과 같습니다.



varname=*value*

값을 변수 이름에 할당합니다.

예를 들어, DEPT라는 글로벌 변수를 설정하려면 다음 명령을 발행하십시오.

선형 프로시저의 경우,

```
SET GLOBAL (DEPT=38
```

논리를 갖는 프로시저의 경우,

```
"SET GLOBAL (DEPT=38"
```

SET GLOBAL 명령에 관한 자세한 정보는 *QMF* 참조서를 참조하십시오.

글로벌 변수를 정의하면 변수를 다시 설정하거나 QMF 세션을 종료할 때까지 해당 글로벌 변수는 정의된 상태를 유지합니다. RESET GLOBAL 명령 사용에 관한 내용을 보려면 *QMF* 참조서를 참조하십시오.

SET GLOBAL: 확장 구문

REXX 이외의 언어(어셈블러, C, COBOL, FORTRAN 또는 PL/I)로 작성된 응용프로그램에서 이들 변수 값을 변경하려면 확장 구문에 SET GLOBAL 명령을 사용해야 합니다. 이 명령의 예제는 147 페이지의 『부록A. 호출 인터페이스 언어의 샘플 코드』에서 적절한 언어에 맞는 샘플 프로그램을 참조하십시오.

SET GLOBAL 확장 구문 명령으로 사용되는 변수 이름의 최대 길이는 17자입니다. 변수 값의 최대 길이는 32 768자입니다.

▶▶—SET GLOBAL—(—| Variable definitions |—————▶▶

Variable definitions:

|—number of varnames—,—varname lengths—,—varnames—,——————|

|—value lengths—,—values—,—value type——————|

number of varnames

요청되는 변수의 수

varname lengths

지정된 각 변수 이름에 대한 길이 목록

변수 이름의 길이는 기억영역에서 글로벌 이름의 실제 길이와 같습니다. 맨 뒤 공백으로 채워지는 18자의 영역이 허용됩니다.

varnames

QMF 변수의 이름 목록

value lengths

변수 값의 길이 목록

다음 규칙은 변수 값에 적용됩니다.

- 제공한 값 길이가 기억영역에서 저장된 값의 길이보다 짧으면 값은 QMF에서 저장된 경우 오른쪽에서 잘립니다.
- 제공한 값 길이가 기억영역에 저장된 값의 길이보다 길면 값은 QMF에서 저장된 경우 값은 인식할 수 없는 문자를 가진 것처럼 나타날 수도 있습니다.
- 정수 길이는 항상 4바이트입니다.

QMF는 기억영역에 있는 값을 사용하고 할당된 길이에 대해 할당된 주소에서 작업을 시작합니다. 길이가 너무 길면 QMF는 이상 종료할 수도 있습니다.

values

변수 값 목록

value type

값을 포함하는 기억영역의 데이터 유형. 이 값은 문자나 정수 중 하나여야 합니다.

REXX 호출 인터페이스에서 SET GLOBAL을 사용할 경우 73 페이지의 『SET GLOBAL』에 나타난 것처럼 SET GLOBAL 명령에 대해 선형 구문만 사용할 수 있습니다. 이 선형 구문에서 글로벌 변수 이름의 최대 길이는 17자이고 변수 값의 최대 길이는 55자입니다.

글로벌 변수 사용에 대한 규칙

- SET GLOBAL 명령에서 변수 이름이 RUN 및 CONVERT 명령에 있을 경우 그 앞에 앰퍼샌드가 사용되지 않습니다.
- QMF 서식은 서식 변수 이름이나 집계 변수 이름에 설정된 글로벌 변수를 인식하지 않습니다.
- QMF 서식은 이름에 물음표가 들어간 글로벌 변수를 인식하지 않습니다.

글로벌 변수 이름 정의에 대한 규칙

- 명령행에서 입력된 경우 글로벌 변수 이름은 17자로 제한되며 호출 인터페이스를 통해 입력된 경우 18자로 제한됩니다. 그러나 SET GLOBAL 명령의 제한으로 인해 17자 이름만이 사용됩니다.
- 글로벌 변수 이름은 숫자를 포함할 수 있으나 글로벌 변수 이름의 첫번째 문자는 숫자가 될 수 없습니다.
- 사전 정의된 QMF 글로벌 변수에 대해 QMF에서 예약된 DSQ로 글로벌 변수를 시작할 수 없습니다.
- 글로벌 변수 이름 중 첫번째 문자는 영문자(A - Z)이거나 다음 특수 문자 중 하나여야 합니다.

¢ ! \$ % & ' () * + , - . / : ; < > [\] ^ _ ` { | } ~

- 글로벌 변수 이름에는 공백 또는 다음 문자 중 어느 것도 포함될 수 없습니다.

* () - + - | : ; " ' < > / . , = &

- 맨 뒤 공백은 글로벌 변수 이름으로 인식되지 않습니다.

values

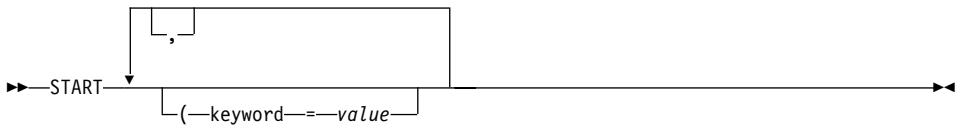
이 명령에서 지정된 시작 명령 키워드에 대한 값 목록

value type

값을 포함하는 기억영역의 데이터 유형. START 명령의 값 유형은 문자여야 합니다.

REXX 호출 인터페이스의 시작 명령 구문

REXX 호출 인터페이스의 경우, START 명령의 구문은 다음과 같습니다.



START 명령 키워드

START 명령에 다음 키워드 중 하나를 지정하십시오.

DSQADPAN

DSQSIROW

DSQALANG

DSQSMODE ²

DSQSBSTG

DSQSPILL

DSQSCMD ²(CMS 및 TSO의 경우)

DSQSPLAN(TSO의 경우)

DSQSDBCS

DSQSPRID(TSO의 경우)

DSQSDBNM

DSQSRSTG(CMS 및 TSO의 경우)

DSQSDBQN(CICS의 경우)

DSQSRUN ²

2. 이 키워드는 SAA 명령 키워드입니다.

DSQSDBQT(CICS의 경우)

DSQSSPQN(CICS의 경우)

DSQSDEBUG

DSQSSUBS(TSO의 경우)

DSQSDCSS(CMS의 경우)

DSQSUSER(CICS/VSE의 경우)

이 키워드는 80 페이지의 표7에서 설명됩니다.

QMF에서는 다음 규약에 따라 START 명령 키워드를 지정할 수 있습니다.

- START 명령에 모든 시작 명령 키워드를 지정할 수 있습니다. CICS를 제외한 QMF에서 지원되는 모든 환경에서 DSQSCMD를 제외하고 DSQSCMD 매개 변수에 의해 명명된 REXX 프로그램에서는 어떠한 키워드도 지정할 수 있습니다. QMF CICS에서 REXX를 지원하지 않으므로 START 명령에 모든 키워드를 지정해야 합니다.
- 키워드를 지정하지 않으면 QMF는 DSQSCMD 키워드에 의해 지정되는 프로그램에 나타나는 대로 START 명령 키워드의 값을 사용합니다. 이 프로그램을 사용하지 않을 경우 QMF는 각 키워드의 기본값을 사용합니다.
- 응용프로그램 또는 초기 프로시저가 특정 환경에서 지원되지 않은 키워드를 지정하면 해당 키워드는 무시됩니다. 이 방법으로, 환경 특정 키워드를 변경하지 않고 여러 QMF 환경에서 실행되는 단일 프로그램을 컴파일할 수 있습니다.

이 키워드에 대한 자세한 정보와 이러한 키워드가 환경에 어떤 영향을 받는지에 대해 자세히 알려면 사용하는 플랫폼에 대한 *Installing and Managing QMF*를 참조하십시오. 80 페이지의 표7에서, 키워드 이름별 위첨자 SAA⁽²⁾는 SAA 시작 명령 키워드를 표시합니다.

응용프로그램의 QMF 명령

표 7. START 명령 키워드, 설명, 기본값

START 명령 키워드	설명	기본값																														
DSQADPAN	DSQDC_DISPLAY_RPT 글로벌 변수를 설정합니다. 이 변수는 조회가 응용프로그램에서 실행될 경우 QMF가 보고서를 표시하는지 여부를 제어합니다. 조회가 실행되는 경우 1값은 보고서를 표시합니다. 0값은 보고서가 표시되지 않음을 지정합니다.	호출 인터페이스의 경우: 1 일괄처리 모드에서 또는 QMF가 DSQQMFE로 대화식으로 시작한 경우: 0																														
DSQALANG	<p>시작하는 세션에 대해 기본 언어를 판별합니다. 이 매개변수에 대한 값은 단일 문자로 된 언어 ID입니다. 이 키워드에 의해 지정된 기본 언어에서 QMF 명령을 입력하거나 지정하십시오. 기본 언어가 영어 이외의 언어일 경우 영어 명령을 입력하려면 QMF 2개 언어 지원을 사용할 수 있습니다(51 페이지의 『제6장 2개 언어 응용프로그램 작성』 참조). 다음 표에서는 이 변수에 유효한 값인 언어 ID의 전체 목록을 보여줍니다.</p> <table border="1"> <thead> <tr> <th>ID</th> <th>언어</th> </tr> </thead> <tbody> <tr> <td>D</td> <td>독일어</td> </tr> <tr> <td>E</td> <td>영어</td> </tr> <tr> <td>F</td> <td>프랑스어</td> </tr> <tr> <td>H</td> <td>한글(한국)</td> </tr> <tr> <td>I</td> <td>이탈리아어</td> </tr> <tr> <td>K</td> <td>간지(일본)</td> </tr> <tr> <td>P</td> <td>브라질 포르투갈어</td> </tr> <tr> <td>Q</td> <td>덴마크어(QMF VSE에서 사용불가능)</td> </tr> <tr> <td>R</td> <td>중국어(S-Chinese)</td> </tr> <tr> <td>S</td> <td>스페인어</td> </tr> <tr> <td>U</td> <td>대문자 영어</td> </tr> <tr> <td>V</td> <td>스웨덴어(QMF VSE에서 사용불가능)</td> </tr> <tr> <td>Y</td> <td>스위스 프랑스어</td> </tr> <tr> <td>Z</td> <td>스위스 독일어</td> </tr> </tbody> </table>	ID	언어	D	독일어	E	영어	F	프랑스어	H	한글(한국)	I	이탈리아어	K	간지(일본)	P	브라질 포르투갈어	Q	덴마크어(QMF VSE에서 사용불가능)	R	중국어(S-Chinese)	S	스페인어	U	대문자 영어	V	스웨덴어(QMF VSE에서 사용불가능)	Y	스위스 프랑스어	Z	스위스 독일어	영어의 경우 E
ID	언어																															
D	독일어																															
E	영어																															
F	프랑스어																															
H	한글(한국)																															
I	이탈리아어																															
K	간지(일본)																															
P	브라질 포르투갈어																															
Q	덴마크어(QMF VSE에서 사용불가능)																															
R	중국어(S-Chinese)																															
S	스페인어																															
U	대문자 영어																															
V	스웨덴어(QMF VSE에서 사용불가능)																															
Y	스위스 프랑스어																															
Z	스위스 독일어																															

표 7. START 명령 키워드, 설명, 기본값 (계속)

START 명령 키워드	설명	기본값
DSQSBSTG	<p>보고서 생성을 위해 사용할 기억영역의 바이트 수를 나타냅니다. CICS에서 같은 주소 공간에 여러 사용자가 있으면 기억영역의 양을 제한해야 하고 이 변수가 지정되었다면 TSO와 CMS에서도 유사하게 사용됩니다.</p> <p>TSO와 CMS에서 둘다를 지정하면 이 키워드는 DSQSRSTG 키워드를 덮어씁니다. 이 키워드를 지정하지 않으면 DSQSRSTG 키워드가 사용됩니다.</p> <p>CICS 사용자를 위한 주의사항: DSQSBSTG는 CICS에 대해 항상 사용됩니다. DSQSRSTG는 CICS에 대해서는 결코 사용되지 않습니다.</p> <p>보고서를 생성하는 데 필요한 기억영역의 최소 합계보다 적은 값으로 DSQSBSTG의 값을 설정하면 QMF는 자동으로 최소 필수 기억영역을 할당합니다. 이 최소값은 환경에 따라 달라집니다. 큰 보고서는 최소 기억영역 양 이상을 필요로 할 수 있습니다.</p>	<p>CICS의 경우: 500 000바이트</p> <p>CMS 또는 TSO의 경우: 0 바이트</p>
DSQSCMD ² (CMS 및 TSO의 경우)	<p>QMF 프로그램 매개변수를 설정하는 REXX 프로그램을 지정합니다.</p> <p>호출 인터페이스 응용프로그램으로부터 QMF가 START 명령을 수신하면 QMF는 이 키워드에 의해 지정된 REXX 프로그램을 호출합니다. 이 REXX 프로그램은 QMF가 START 명령에서 지정되지 않는 키워드에 대한 기본값으로 사용하는 QMF 프로그램 매개변수에 대한 값을 제공합니다.</p> <p>START (DSQSCMD=yourprogram)</p> <p>CICS 사용자를 위한 주의사항: QMF CICS에서는 REXX를 지원하지 않으므로 DSQSCMD는 CICS에서 지원되지 않습니다. CICS에서 호출 인터페이스를 사용하는 QMF를 시작하고 QMF 프로그램 매개변수를 설정하려면 START 명령에 이들 키워드를 지정해야 합니다.</p>	DSQSCMDE
DSQSDBCS	<p>터미널이 DBCS를 지원하지 않을 경우 QMF가 2바이트 문자를 허용하는지 여부를 결정합니다. 값은 YES와 NO입니다.</p> <p>비 DBCS 터미널 또는 DBCS 데이터를 인쇄하는 QMF 일괄처리 작업을 실행하여 2바이트 문자 세트 데이터를 인쇄하려면 값을 YES로 설정하십시오. 그렇지 않으면 값은 NO가 됩니다.</p>	NO

응용프로그램의 QMF 명령

표 7. START 명령 키워드, 설명, 기본값 (계속)

START 명령 키워드	설명	기본값
DSQSDBNM	QMF 세션을 시작할 때 연결된 위치를 지정합니다. 널 값을 설정하면 QMF가 기본 데이터베이스와 연결됩니다(원격 작업 단위 없이 정상적으로 연결하는 데이터베이스). 통신 데이터베이스가 설치되지 않았고 널(null) 또는 기본 데이터베이스의 이름 이외의 값을 지정할 경우 오류가 발생합니다.	NULL
DSQSDBN (CICS의 경우)	CICS 기억영역이 QMF 추적 데이터에 대해 사용되도록 지정합니다. 이름은 DSQSDBQT에 의해 선택된 CICS 대기열 유형에 대한 CICS 이름 세부사항을 준수합니다. CICS 이름 세부사항에 대한 자세한 내용은 사용자 시스템의 CICS 응용 프로그래밍 안내서를 참조하십시오.	DSQD
DSQSDBQT(CICS의 경우)	QMF 추적 데이터의 경우 사용될 CICS 기억영역의 유형을 지정합니다. 값은 다음과 같습니다. TD CICS 임시 데이터를 사용합니다. TS CICS 보조 임시 기억영역을 사용합니다. QMF는 많은 양의 추적 데이터를 생성할 수 있기 때문에 임시 기억영역을 지정할 때에는 주의를 기울여야 합니다.	TD
DSQSDBUG	QMF 초기화 동안 제품 추적이 발생하는지 여부를 지정합니다. 값은 다음과 같습니다. ALL 가장 상세한 QMF 추적을 지정합니다. NONE 최소의 QMF 추적을 지정합니다.	NONE
DSQSDCSS(CMS의 경우)	QMF 실행 모듈을 포함하는 DCSS(비연속 공유 세그먼트)의 이름을 지정합니다.	QMF710 <i>n</i> . 여기서 <i>n</i> 은 자국어 ID입니다. 영어의 경우 기본값은 QMF710E입니다.
DSQSIROW	RUN QUERY, IMPORT DATA 또는 DISPLAY 명령 데이터의 첫 화면을 표시하기 전에 QMF에서 폐치하는 행 수를 나타냅니다.	100

표 7. START 명령 키워드, 설명, 기본값 (계속)

START 명령 키워드	설명	기본값
DSQSMODE ²	어떤 모드에서 작업하는지 QMF에 알립니다. I 대화식 모드를 지정합니다. B 일괄처리 모드를 지정합니다. DSQSMODE의 값이 B이면 QMF가 배경 작업에서 실행되도록 패널 표시가 억제됩니다.	B(일괄처리)
DSQSPILL	QMF가 스푼 파일을 사용하는지 데이터 세트를 사용하는지 여부를 지정합니다. 가능한 값은 YES 또는 NO입니다.	CICS의 경우: NO CMS 및 TSO의 경우: YES
DSQSPLAN(TSO의 경우)	QMF에 할당된 DB2 응용프로그램 플랜 ID를 지정합니다.	QMF710
DSQSPRID(TSO의 경우)	Q.PROFILES에서 적절한 행을 선택하고 Q.ERROR_LOG 항목을 규정하기 위해 TSO 로그온 ID를 사용할지 1차 권한 ID를 사용할지 지정합니다. 허용되는 값은 PRIMEID 또는 TSOID입니다.	PRIMEID
DSQSRSTG(CMS 및 TSO의 경우)	응용프로그램과 응용프로그램에 의해 호출되는 다른 응용프로그램에 대해 예약할 가상 기억영역의 바이트 수를 결정합니다. QMF 환경에서 응용프로그램을 수행하려는 경우 이 매개변수를 사용하십시오. 응용프로그램에 대해 기억영역을 예약하지 않으면 QMF는 큰 보고서를 작성하기 위해 사용가능한 모든 가상 기억영역을 사용할 수 있습니다.	Zero(0)
DSQSRUN ²	QMF가 시작된 후에 실행될 QMF 초기 프로시저의 이름을 지정합니다. 초기 프로시저는 호출 인터페이스를 통해 단 한번만 실행됩니다. 이 프로시저에서, 사용자의 세션을 조정하기 위해 글로벌 변수와 프로파일 변수를 설정하는 명령을 추가할 수 있습니다. 응용프로그램이 QMF를 대화식으로 실행하려 하거나 일괄처리 조작을 위한 QMF 세션을 준비할 경우 이 프로시저는 QMF에 사용자를 배치할 수 있습니다.	NULL
DSQSSPQN(CICS의 경우)	QMF 스푼 데이터에 대해 사용되는 CICS 임시 기억영역 대기열의 이름을 지정합니다. 프로그램 매개변수 DSQSPILL가 YES의 값을 가질 경우 이 스푼 영역에 보고서 데이터가 포함됩니다.	DSQSid. 여기서 id는 CICS 터미널 ID입니다.
DSQSSUBS(TSO의 경우)	QMF가 설치된 DB2 서브시스템의 ID를 지정합니다.	DSN

응용프로그램의 QMF 명령

표 7. START 명령 키워드, 설명, 기본값 (계속)

START 명령 키워드	설명	기본값
DSQSUSER (CICS/VSE의 경우)	CONNECT 명령에 SQL/DS 권한 ID와 암호를 지정합니다. DSQSUSER 키워드를 지정하려면, 다음을 입력하십시오. <i>DSQSUSER=SQL ID/password</i> 여기서, SQL ID는 QMF를 시작하는 사용자에게 대한 SQL/DS™ 권한 ID입니다.	시스템 카탈로그에 정의된 3바이트 VSE 운영자 ID 및 암호. DSQSMODE=B로 QMF를 시작할 경우, 지정해야 합니다.

명령 동의어 사용

QMF는 QMF 명령과 비슷한 명령 동의어를 작성하게 합니다. 명령 동의어는 매우 용통성이 있으며 일반 사용자에게 매우 유용합니다. 예를 들어, 명령 동의어는 명령의 기능을 실행하거나 응용프로그램을 시작합니다. QMF 사용자가 명령 동의어를 액세스할 수 있게 하려면 명령 동의어를 하나 이상의 명령 동의어 표에 입력해야 합니다. 사용자가 명령 동의어를 발행하면 QMF는 사용자의 응용프로그램을 시작하는 TSO, RUN, CICS 또는 CMS 명령을 실행합니다.

명령 동의어 작성

명령 동의어를 작성하려면, 다음을 수행하십시오.

1. 설명적인 명령을 작성합니다.

QMF 명령은 명령어 오브젝트 형식을 따릅니다. 모든 명령은 명령어(조치 언어)이고 많은 명령 다음에는 오브젝트(설명적인 명령어)가 옵니다. 예를 들어, CONVERT QUERY가 명령어 오브젝트 명령인 반면에 END는 명령어만으로 된 명령입니다.

기존 QMF 명령과 동일한 명령어를 사용하여 명령 동의어를 작성할 수 있습니다. 이 경우, QMF 명령을 앞에 두고 원래 QMF 명령을 사용할 수 있습니다. QMF 명령에 관한 내용을 보려면 *QMF* 참조서를 참조하십시오.

예를 들어, 주별 매출액이 입력되었는지를 보기 위해 프로시저가 보고서를 실행한다고 가정합니다. 이번 주에 대한 데이터가 빠졌으면 프로시저는 표 편집기를 호출하여 표에 최근 정보를 추가합니다. 프로시저 이름에 관계없이 명령

동의어에 설명적 개념이 포함되어 있기를 원할 수 있습니다. UPDATE SALES와 같이 명령어 오브젝트 쌍을 선택할 수 있습니다.

명령 동의어에 매개변수나 옵션이 필요하다면, 대체 변수 &ALL.을 사용할 수 있습니다.

2. 적절한 명령 동의어 표를 새 명령 동의어로 갱신합니다.

사용하려는 명령 동의어 표의 이름을 알아야 합니다.

DSQAP_SYNONYM_TBL 변수에는 사용자 각각이 단축 명령 표의 이름이 들어 있습니다.

데이터베이스 관리자는 명령 동의어 표에 액세스할 수 있습니다. 개인적 용도로 명령 동의어를 작성하려면 명령 동의어를 추가하도록 정의된 보기를 가질 수도 있습니다.

명령 동의어 표에는 명령 동의어 UPDATE SALES에 대한 다음과 같은 정보가 들어 있습니다.

```

ADD                                Q.COMMAND_SYNONYMS

VERB. . . . . ( UPDATE                )
OBJECT. . . . . ( SALES                  )
SYNONYM_DEFINITION. . ( RUN PROC WEEKLY_SALES
REMARKS . . . . . ( 주별 매출을 검사하는 프로시저>
                                                    1 -4/4
                                                    >
    
```

ADD 기능 키를 누르면 QMF는 이 명령 동의어를 표에 추가합니다. 그러나 명령을 사용하려면 먼저 QMF에 다시 연결해야 합니다.

3. 필요한 경우, 프로파일을 갱신합니다.

새 표 또는 보기에 이 명령 동의어를 추가했으면 사용자 프로파일에 새 표 또는 보기의 이름을 추가하십시오.

4. QMF 세션을 종료합니다.

새로운 QMF 세션을 시작할 때까지 QMF는 명령 동의어와 프로파일 표에 수행된 변경을 인식하지 않습니다.

SAA RUN QUERY 보고서 소형 세션

QMF 보고서를 생성하는 응용프로그램을 작성할 때에는 보고서 소형 세션을 사용하여 QMF에 사용자의 액세스를 제한할 수 있습니다. 보고서 소형 세션에서, QMF

는 보고서가 화면에 나타나는 동안 사용자가 발행할 수 있는 명령을 제한합니다. 보고서 소형 세션에서 유효하거나 유효하지 않은 명령은 87 페이지의 표8 및 87 페이지의 표9에 정리되어 있습니다.

보고서 소형 세션은 중첩된 세션(세션 내부의 세션)으로 작동됩니다. 소형 세션에서 초기 QMF 세션은 그대로 남아 있으나 보고서가 화면에 나타나는 동안 일시적으로 사용할 수 없게 됩니다. 소형 세션은 END 명령을 발행(또는 종료 종료 키를 누름)할 때까지 사용중인 현재 세션이 됩니다. 소형 세션을 종료하면 응용프로그램 작성법에 따라 초기 QMF 세션이나 호출 응용프로그램으로 리턴됩니다. 보고서 소형 세션이 종료될 때까지 응용프로그램은 하위 명령 발행을 계속할 수 없습니다.

QMF 글로벌 변수 DSQDC_DISPLAY_RPT는 실제로 QMF에서 보고서 소형 세션의 시작 여부를 결정합니다. 이는 DSQDC_DISPLAY_RPT가 조회를 실행한 후 QMF에서 보고서 표시 여부를 결정하기 때문입니다(보고서를 표시할 경우 1, 표시하지 않을 경우 0으로 설정).

호출 인터페이스를 사용하여 QMF를 실행할 경우

- 글로벌 변수 DSQDC_DISPLAY_RPT의 기본값은 1입니다(QMF가 DSQQMFE 로 시작되면 대화식 또는 일괄처리 모드인지에 관계없이 글로벌 변수의 기본값은 0입니다).
- 조회를 실행하는 프로시저나 응용프로그램을 실행하면 QMF는 보고서 소형 세션을 시작합니다. QMF는 이 세션에서 조회 결과 보고서를 표시합니다.
- 프로시저나 응용프로그램이 조회를 실행하지 않거나 SQL 패널로부터 조회를 실행하면 QMF는 보고서 소형 세션을 시작하지 않습니다.

QMF에서 보고서 소형 세션을 시작하지 않게 하려면, 다음 중 하나를 수행하십시오.

- DSQDC_DISPLAY_RPT의 값을 0으로 변경합니다.
- 호출 인터페이스로부터 QMF를 시작할 경우 DSQADPAN 매개변수를 0에 설정합니다.

글로벌 변수에 대해서는 73 페이지의 『SET GLOBAL』을 참조하십시오.

보고서 소형 세션에서, 다음 명령과 이들 명령의 동의어를 발행할 수 있습니다(제한사항은 괄호 안에 표시됨).

표 8. 소형 세션에서 유효한 명령

BACKWARD	FORWARD	RETRIEVE
BOTTOM	GET GLOBAL	RIGHT
CANCEL(팝업 창이 사용중인 경우)	HELP	SAVE(데이터)
CICS	INTERACT	SET(프로파일, 글로벌)
CMS	ISPF	SHOW(보고서, 차트)
DISPLAY(보고서, 차트)	LEFT	SWITCH(도움말이 사용중인 경우)
END	MESSAGE	TOP
ENTER	PRINT(보고서, 차트)	TSO
	QMF	

표9에는 소형 세션에서 유효하지 않은 명령 목록이 나와 있습니다.

표 9. 소형 세션에서 유효하지 않은 명령

ADD	ERASE	REDUCE
CANCEL	EXIT	REFRESH
CHANGE	EXPORT	RESET GLOBAL
CHECK	EXTRACT	RESET(조회, 프로시저, 서식)
CLEAR	GETQMF	RUN
CONNECT	IMPORT	SAVE
CONVERT	INSERT	SEARCH
DELETE	INTERACT	SHOW
DESCRIBE	IRM	SORT
DISPLAY(조회, 프로시저, 프로파일, 서식)	LIST	SPECIFY
DRAW	NEXT	START
EDIT	PREVIOUS	SWITCH
ENLARGE	PRINT(조회, 프로시저, 프로파일, 서식)	

제한된 명령을 발행하는 exec, CLIST 또는 프로시저를 실행하는 경우 QMF는 오류 메시지를 리턴시킵니다.

제8장 QMF 오브젝트 가져오기 및 내보내기

QMF 환경의 외부에서 QMF 오브젝트를 사용하는 응용프로그램을 작성할 수 있습니다. QMF 환경 외부에 QMF 오브젝트를 배치하려면 QMF EXPORT 및 IMPORT 명령을 사용합니다.

다음 오브젝트를 내보낼 수 있습니다.

차트(chart)

데이터

서식(form)

프로시저(proc)

조회(query)

보고서

표(table)

오브젝트를 내보낼 경우 QMF는 오브젝트를 외부 형식으로 전환하고 그것을 파일, 데이터 세트 또는 CICS 데이터 대기열에 추가합니다. QMF 오브젝트의 외부 형식은 QMF 응용프로그램 개발에 있어 강력한 요소입니다. IMPORT 명령은 파일, 데이터 세트 또는 CICS 데이터 대기열에서 외부 형식을 읽고 오브젝트를 QMF 임시 기억영역이나 데이터베이스 중 하나에 추가합니다(명령 발행 방법에 따라 달라짐).

QMF 또는 IXF 형식 중 하나로 데이터 및 표 오브젝트를 내보낼 수 있습니다. 서식, 프롬프트 조회, 보고서 오브젝트 형식은 코드화 형식이라 불리는 좀더 복잡한 형식입니다. 차트는 그래픽 데이터 형식(GDF)인 GDDM 형식으로 내보내집니다.

이 장에서는 모든 QMF 내보내기 형식을 설명하고 응용프로그램에서 그것을 사용하는 방법을 보여줍니다. 251 페이지의 『부록B. 내보내기/가져오기 형식』은 데이

QMF 오브젝트 가져오기 및 내보내기

터의 QMF 형식을 설명하고 코드화된 형식 오브젝트의 표와 필드 번호를 정의합니다. IXF 형식에 관한 내용을 보려면 265 페이지의 『부록C. 통합 교환 형식(IXF)』을 참조하십시오.

CICS 사용자: IMPORT 또는 EXPORT 명령을 사용하는 응용프로그램을 작성할 경우 138 페이지의 『CICS 대기열 사용 규칙 및 고려사항』을 읽어 보십시오.

IMPORT 및 EXPORT 명령의 구문을 보려면 *QMF 참조서*를 참조하고 QMF 오브젝트를 가져오고 내보내는 데 필요한 내용을 보려면 온라인 도움말을 참조하십시오.

내보낸 파일, 데이터 세트 또는 CICS 데이터 대기열로 할 수 있는 작업

가져오기/내보내기 기능은 다음과 같은 역할을 합니다.

- 조회 결과를 응용프로그램에 제공합니다.
많은 응용프로그램의 목적은 QMF 조회에 의해 생성된 데이터를 사용하는 것입니다. 데이터를 데이터베이스에서 응용프로그램으로 가져오려면 QMF EXPORT 명령을 사용하십시오.
- 응용프로그램에서 오브젝트를 작성하고 QMF에서 이를 사용합니다.
오브젝트에 대한 적절한 형식을 사용하여 QMF 환경 외부에서 오브젝트를 작성할 수 있습니다. 오브젝트를 포함하는 파일, 데이터 세트 또는 CICS 데이터 대기열을 QMF로 가져올 경우 새 QMF 오브젝트가 작성됩니다.
보고서와 차트를 QMF로 가져올 수 없습니다.
- 데이터베이스에 비 QMF 오브젝트를 저장합니다.
프로시저 또는 조회 오브젝트로서 오브젝트를 가져올 경우 QMF는 오브젝트를 QMF 환경에 그대로 가져옵니다. 그것은 추가 레코드 또는 필드를 가져온 파일에 삽입하지 않습니다. 레코드의 길이가 79바이트 이하인 모든 프로그램과 파일을 가져올 수 있습니다.
- 오브젝트를 다른 환경 또는 제품에 사용할 수 있도록 만듭니다.

주의:

시스템이나 환경 사이에서 전송되는 내보낸 오브젝트는 내보낸 오브젝트를 변경하거나 손상시킬 수 있는 변환 위험에 노출되어 있습니다. EBCDIC과 ASCII 시스템 사이 또는 서로 다른 NLF 환경 사이와 같이 서로 다른 CCSIS 또는 문자 세트로 실행되는 환경 사이에서 내보낸 오브젝트를 전송하는 것은 바람직하지 않습니다.

CONVERT QUERY 명령을 사용하여 프롬프트 조회 또는 QBE 조회를 다른 제품에서 내보내고 사용할 수 있는 SQL 조회로 변환시킬 수 있습니다. CONVERT 명령에 관한 자세한 내용은 QMF 참조서를 참조하십시오.

다음 위치에서 QMF 오브젝트를 전송할 수 있습니다.

- VM에서 CMS 세션간에
- CICS 외부 파티션 임시 데이터 대기열을 사용하여 TSO 또는 고유 OS/390 일괄처리의 QMF 및 CICS하의 QMF 사이에서
- SENDFILE 네트워크에서
- 데이터베이스 외부에 오브젝트와 데이터를 저장합니다.

예를 들어, 프로그램 중간에 외부 프로그램이 처리할 수 있도록 하기 위해 데이터를 내보낼 수 있습니다.

- 2개 언어 응용프로그램을 작성합니다.

기본 언어로 QMF 서식을 작성할 수 있고 EXPORT 명령의 LANGUAGE= 선택을 사용하여 영어로 번역할 수 있습니다. IMPORT 및 EXPORT 명령상에 있는 LANGUAGE= 선택을 사용하여 영어 서식을 기본 영어로 번역할 수 있습니다.

데이터 내보내기 및 저장

EXPORT DATA와 SAVE DATA 사이의 차이는 오브젝트가 저장되는 위치와 방법으로 이것은 결과로 수행할 수 있는 작업에 영향을 줍니다.

- 데이터 오브젝트를 내보내면 파일, 데이터 세트 또는 CICS 데이터 대기열이 생성됩니다. 각각을 순서대로 읽고, 변경하고, 인쇄할 수 있으나, QMF 응용프로그램 또는 외부의 다른 응용프로그램을 통해 이러한 모든 작업을 수행할 수 있습니다.

QMF 오브젝트 가져오기 및 내보내기

- SAVE DATA 명령은 데이터베이스 표를 생성합니다. 저장된 데이터를 사용하여 수행하는 모든 조치는 데이터베이스를 통해 실행됩니다.

데이터 및 표 오브젝트

조치를 실행할 경우 QMF는 데이터 오브젝트나 보고서 오브젝트 중 하나로 내보낼 수 있는 보고서에서 결과를 표시합니다. 보고서 오브젝트를 내보낼 경우 오브젝트는 서식 오브젝트에 지정된 데이터의 형식을 유지합니다. HTML 보고서로 보고서 오브젝트를 내보내면 이는 적절한 HTML 3.0 코딩으로 패키징됩니다. 보고서를 웹 서버에 넣어 월드 와이드 웹에 표시되게 할 수 있습니다. QMF 데이터와 표 오브젝트는 원시 데이터 서식으로 내보내집니다. 보고서 오브젝트에 관한 자세한 내용은 127 페이지의 『보고서 오브젝트』를 참조하십시오.

표 서식으로 표시할 원시 데이터는 데이터 오브젝트로서 임시 기억영역에 저장됩니다. 데이터베이스에 저장된 관계형 표 및 보기는 표 오브젝트라 불립니다. 임시 기억영역 내의 표(DATA) 및 데이터베이스에 저장된 표(TABLE)의 내보낸 형식은 동일합니다. 데이터로서 내보낸 오브젝트는 표로 가져올 수 있고 표로서 내보낸 오브젝트로 데이터를 가져올 수 있습니다.

데이터 및 표 오브젝트를 QMF 형식이나 통합된 교환 형식(IXF)으로 내보냅니다.

원하는 내보내기 형식을 QMF에 알리기 위해 EXPORT 명령에 DATAFORMAT=QMF 또는 DATAFORMAT=IXF 중 하나를 지정할 수 있습니다. QMF 형식이 기본값입니다. QMF 형식에 대해서는 251 페이지의 『데이터에 대한 QMF 형식』에서 설명됩니다.

IXF에는 두 개의 형식인 2진과 문자 형식이 있는데 이는 96 페이지의 『2진 대 문자』에서 설명됩니다. IXF 형식은 265 페이지의 『부록C. 통합 교환 형식(IXF)』에서 설명됩니다.

QMF 또는 IXF 형식을 지정하고 필요한 데이터를 포함하는 파일, 데이터 세트 또는 CICS 데이터 대기열을 가져옴으로써 &file에 자신의 표를 작성할 수 있습니다. 필요한 필드를 적절하게 포함시키고 자신의 데이터를 추가합니다. 그런 다음 이 파일, 데이터 세트 또는 CICS 데이터 대기열을 표 오브젝트로서 QMF로 가져옴

니다. 다음은 파일, 데이터 세트 또는 CICS 데이터 대기열을 표 오브젝트로서 데이터베이스에 가져오는 명령의 예제입니다.

```
IMPORT TABLE MYTABLE FROM MYDATA
```

EXPORT 및 IMPORT 명령에 관한 자세한 내용은 QMF 참조서를 참조하십시오. CICS/VSE 사용자는 138 페이지의 『CICS 대기열 사용 규칙 및 고려사항』을 읽어 보십시오.

QMF 형식의 데이터 오브젝트 해석: 예제

데이터 레코드의 길이가 있으면 헤더 레코드의 길이를 계산할 수 있습니다. 이 예제에서 각 데이터 레코드는 23바이트 길이입니다. 251 페이지의 『데이터에 대한 QMF 형식』은 처음 레벨과 수의 정보를 포함하는 첫 12바이트를 설명합니다. 데이터의 각 열에는 24바이트가 있고 세 개의 열이 있습니다. 따라서 이 세 열 데이터 오브젝트의 경우 헤더는 84바이트입니다.

$$(12 + (24 \times 3) = 84)$$

Q.STAFF에서 다음 데이터를 내보내려면 다음과 같습니다.

ID	NAME	COMM
10	SANDERS	-
20	PERNAL	612.45

다음 표를 사용하여 각 열의 너비를 계산할 수 있습니다.

표 10. 열 너비 계산

열 이름	데이터 유형	데이터 유형 너비(헤더의 길이)	열의 너비
ID	SMALLINT	2	2 + 2 = 4
NAME	VARCHAR	9	2 + 2 + 9 = 13
COMM	DECIMAL (7,2)	7	(7 + 1)/2 + 2 = 6
데이터 레코드의 길이			23

각 헤더 레코드는 23바이트로 데이터 레코드와 길이가 같습니다. 이 84바이트는 네 개의 23바이트 헤더 레코드로 분포되며 마지막 레코드는 공백으로 채워집니다.

QMF 오브젝트 가져오기 및 내보내기

그림15에서는 보고서 헤더 및 16진법을 보여줍니다. 반전된 유형의 수는 그림 다음에 주가 있음을 나타냅니다.

```

1   R E L   1 . 0           I D
   D9 C5 D3 40 F1 4B F0 40 0004 0003 C9 C4 40 40 40 40 40 40 40 40
   1           2 3 4
2   40 40 40 40 40 40 40 01F4 0002 D5 00 D5 C1 D4 C5 40 40 40 40 40 40
   5 6 7
3   40 40 40 40 40 40 40 01C0 0009 E8 00 C3 D6 D4 D4 40 40 40 40 40
   Y           C O M M
4   40 40 40 40 40 40 40 01E4 07 02 E8 00 40 40 40 40 40 40 40
   Y

```

그림 15. QMF 형식으로 내보낸 데이터 오브젝트의 샘플 헤더 레코드. 40은 공백 문자의 16진법 코드입니다.

그림16에서는 보고서 및 해당 데이터를 16진법 표현을 보여줍니다. 바이트 위치의 의미에 대해서는 251 페이지의 『데이터에 대한 QMF 형식』을 참조하십시오.

```

1   00 00 00 0A 00 00 00 07 E2 C1 D5 C4 C5 D9 E2 00 00 FF FF 00 00 00 40 40
   8           9           10
2   00 00 00 14 00 00 00 06 D7 C5 D9 D5 C1 D3 00 00 00 00 00 00 61 24 5C
   20           P E R N A L

```

그림 16. QMF 형식으로 내보낸 데이터 오브젝트의 샘플 데이터 레코드

1 REL 1.0

오브젝트 형식 레벨: 1.0

오브젝트 형식 레벨은 이 오브젝트가 사용하고 있는 오브젝트 형식의 버전을 QMF에 알려줍니다. QMF 오브젝트 형식이 변경될 때마다 레벨 수도 변경됩니다. 오브젝트 형식이 모든 새 릴리스마다 변경되는 것은 아닙니다.

2 X'0004'

헤더 레코드의 수: 4

3 X'0003'

데이터 열의 수: 3

4 X'C9 C4'

열 이름: ID

5 X'1F4'

데이터 유형: SMALLINT

6 X'0002'

열 너비: 2

7 X'D5'

널(NULL)은 허용됩니다. N은 no를 의미합니다.

8 X'0A'

첫번째 데이터 레코드의 첫번째 열 값: 10

9 X'07'

첫번째 데이터 레코드의 두 번째 열에서 이름의 길이: 7

10 X'FFFF'

표시기 정보: 열은 널 값을 포함합니다.

데이터나 표 오브젝트가 반출되는 경우 생성된 파일, 데이터 세트 또는 CICS 데이터 대기열에 관한 자세한 내용은 137 페이지의 『외부화 QMF 오브젝트의 세부사항』을 참조하십시오.

IXF 형식 예제를 보려면 265 페이지의 『부록C. 통합 교환 형식(IXF)』을 참조하십시오.

데이터 및 표 오브젝트의 내보내기가져오기 규칙과 정보

다음은 데이터 또는 표 오브젝트를 가져오고 내보낼 때의 일반적 고려사항입니다.

파일, 데이터 세트 또는 CICS 데이터 대기열은 할당 상태로 있습니다.

QMF 임시 기억영역에 데이터를 저장하고 화면에 보고서를 표시하기 위해 QMF IMPORT DATA 명령이 나타납니다. 실제로 데이터의 일부만이 저장되고 표시됩니다. 파일, 데이터 세트 또는 CICS 데이터 대기열은 열린 상태이고 QMF에 할당됩니다. 사용자가 파일, 데이터 세트 또는 CICS 데이터 대기열을 통해 화면이 동할 때 QMF는 레코드를 읽습니다.

이 연결은 데이터 오브젝트가 대체되거나 재설정되거나 QMF가 모든 레코드를 읽을 때까지 유지됩니다. 이 시점에서 파일, 데이터 세트 또는 CICS 데이터 대기열은 닫히고 더 이상 QMF에 할당된 것으로 여겨지지 않습니다. 이것은 응용프로그램이 IMPORT DATA 명령으로 QMF에 할당된 파일, 데이터 세트 또는 CICS 데이터 대기열을 삭제하거나 변경하려고 시도해서는 안 된다는 점을 의미합니다. 응

QMF 오브젝트 가져오기 및 내보내기

응용프로그램은 읽어오던 데이터 세트를 변경하거나 삭제하려고 하기 전에 다른 데이터 소스 사용을 시작하거나 QMF 임시 데이터 기억영역(RESET DATA)을 비워야 합니다.

IMPORT 명령이 실행되는 동안 QMF는 파일, 데이터 세트 또는 CICS 데이터 대기열이 읽혀지는 중에는 이러한 항목을 잠그지 않습니다. 읽혀지는 동안에는 파일, 데이터 세트 또는 CICS 데이터 대기열이 변경되는 것을 금지하는 단계를 가지지 않습니다. QMF가 읽는 것을 마치기 전에 파일, 데이터 세트 또는 CICS 데이터 대기열이 어떤 방법으로든 지워지거나 변경될 경우 결과는 예측할 수 없고 시스템 오류가 발생합니다.

QMF가 오브젝트를 완료해야 하며 데이터 오브젝트용 공간이 충분히 없을 경우에는 불완전한 데이터 프롬프트가 발생할 수 있습니다. 예를 들어 같은 파일, 데이터 세트 또는 CICS 데이터 대기열로 오브젝트 내보내기를 요청할 경우, QMF는 데이터 오브젝트를 완료합니다. 이 상황은 현재 EXPORT 명령에 명명된 같은 파일, 데이터 세트 또는 CICS 데이터 대기열로부터 이전에 IMPORT DATA 명령을 이전에 실행했음을 의미합니다. 불완전한 데이터 프롬프트에 관한 자세한 내용 및 수행할 조치에 대해서는 다음을 참조하십시오.

2진 대 문자

QMF 형식 또는 IXF(OUTPUTMODE=BINARY) 형식의 2진 서식을 사용하여 데이터 또는 표 오브젝트를 내보내면 데이터는 원시 2진 서식으로 존재합니다. 그러나 IXF(OUTPUTMODE=CHARACTER) 문자 서식을 사용할 경우 내보낸 데이터는 EBCDIC 서식으로 존재합니다. 서식, 보고서, 프로시저 및 SQL 조회 오브젝트의 내보낸 데이터 역시 EBCDIC 서식으로 존재합니다.

보통 PL/I, COBOL 및 어셈블러와 같은 언어로 쓰여진 응용프로그램은 문자 데이터보다 더 빠르고 더 효과적으로 2진 데이터를 읽고 처리할 수 있습니다. 한 IBM 제품에서 다른 제품으로 데이터를 교환하는 작업은 2진 형식일 때 좀더 효과적으로 이루어집니다. 그러나 응용프로그램이 REXX로 작성되었거나 편집기로 데이터를 처리할 경우에는 EBCDIC(문자) 데이터가 좀더 효과적일 것입니다.

오류

QMF는 파일, 데이터 세트 또는 CICS 데이터 대기열에서 데이터를 가져온 후에 보고서 패널 및 확정 메시지를 표시합니다. 파일, 데이터 세트 또는 CICS 데이터

대기열에 형식 오류가 있을 경우, QMF는 보고서 패널을 표시하지 않습니다. 대신에, QMF가 IMPORT 명령을 처리하기 전에 현재 패널이었던 오브젝트 패널에 오류 메시지를 표시합니다. 그러나 현재 오브젝트 패널이 보고서 패널이고 QMF가 가져온 데이터에서 오류를 발견하면 QMF는 시작 패널과 오류 메시지를 표시합니다.

서식 오브젝트와는 달리, 데이터 또는 표 오브젝트를 가져오면 입력 파일, 데이터 세트 또는 CICS 데이터 대기열의 형식은 같은 오브젝트가 EXPORT DATA 또는 EXPORT TABLE 명령을 사용하여 내보낼 경우 생성될 출력 파일, 데이터 세트 또는 CICS 데이터 대기열의 서식과 정확하게 같아야 합니다.

프로시저 및 SQL 조회

이 오브젝트를 나타내는 파일, 데이터 세트 또는 CICS 데이터 대기열의 형식은 모든 오브젝트 중에 가장 간단합니다. 파일, 데이터 세트 또는 CICS 데이터 대기열에서 각 레코드는 기본적으로 화면에 나타나는 그대로의 한 행의 이미지입니다 (고정된 레코드의 길이는 79바이트임).

SQL 조회는 다음과 같습니다.

SQL 조회	수정	행	1
--------	----	---	---

```
SELECT *
FROM Q.STAFF
```

다음은 외부 형식을 가진 조회입니다.

```
*** Top of File ***
SELECT *
FROM Q.STAFF
```

```
*** End of File ***
```

레코드 형식이 간단하기 때문에 QMF 외부에서 SQL 조회나 프로시저를 작성하거나 편집하는 것은 매우 간단합니다. SQL 조회 또는 프로시저는 79바이트 조회 또는 프로시저 레코드를 포함하는 고정 길이 파일, 데이터 세트 또는 CICS 데이터 대기열로 되어 있습니다.

QMF 오브젝트 가져오기 및 내보내기

다. 결과 파일, 데이터 세트 또는 CICS 데이터 대기열을 가져오면 조회 또는 프로시저는 실행할 준비가 된 QMF 임시 기억영역에 존재합니다.

차트 오브젝트

QMF 환경의 외부에서 처리하도록 차트 오브젝트를 내보낼 수 있습니다. 차트는 데이터베이스에 QMF 오브젝트로서 저장될 수 없으며 데이터베이스로부터 검색될 수 없습니다. 차트를 QMF로 가져올 수 없습니다.

QMF는 차트 오브젝트를 내보낼 때 보고서로부터의 데이터를 그래픽 데이터 형식(GDF)으로 전환합니다. GDF(GDDM 형식)은 기존의 데이터 교환 표준입니다. 예를 들면, GDDM 유틸리티를 사용하여 내보낸 차트 데이터를 인쇄하거나 문서 스크립트 파일에 이를 포함시킬 수 있습니다. GDF 형식에 관한 자세한 내용은 GDDM 응용프로그램 프로그래밍 안내서를 참조하십시오.

GDF 형식의 파일 또는 데이터 세트를 사용하는 것과 똑같은 방식으로 내보낸 차트 오브젝트를 사용할 수 있습니다. 예를 들면 다음과 같습니다.

- 문서 작성 기능(DCF)을 사용하면 응용프로그램은 QMF 보고서(인쇄되거나 내보낸 보고서 사용)와 QMF 차트(내보낸 차트 사용)를 결합할 수 있고 형식이 지정된 정보를 프린터로 보낼 수 있습니다.
- 그래픽 편집기를 GGXA 그래픽 편집기로서 사용하면 응용프로그램은 내보낸 QMF 차트를 좀더 변경하고 개선할 수 있습니다.

코드화 오브젝트

서식 및 프롬프트 조회를 코드화된 형식으로 내보내고 가져오며 이 형식은 오브젝트를 표 구조로 변환하는 형식입니다. 코드화된 형식은 오브젝트의 특정 부분을 좀더 쉽게 조작하도록 도와줍니다. 보고서 오브젝트 역시 코드화된 형식으로 내보내지만 보고서를 가져올 수는 없습니다.

서식, 보고서 또는 프롬프트 조회의 코드화 형식(관계형 또는 엔티티 관계형)은 다음 레코드로 구성됩니다.

- 고정 형식 레코드: 헤더 레코드(H)(99페이지 참조)
- 가변 형식 레코드

- 데이터 값 레코드(V)(104페이지 참조)
- 데이터 표 설명 레코드(T)(105페이지 참조)
- 표 행 레코드(R)(109페이지 참조)
- 오브젝트 레코드의 종료(E)(111페이지 참조)

별표(*)가 붙은 응용프로그램 데이터 레코드는 내보낸 파일에 오브젝트와 관계있는 정보와 주석을 저장하는 응용프로그램에 의해 사용됩니다. 자세한 내용은 111 페이지의 『응용프로그램 데이터 레코드(*)』를 참조하십시오.

내보낸 보고서는 위의 레코드와 함께 다음 레코드를 포함할 수 있습니다.

- 보고서 행 레코드(L)(130페이지 참조)
- 데이터 연속 레코드(C)(132페이지 참조)

내보낸 파일, 데이터 세트 또는 CICS 데이터 대기열에 대한 세부사항은 137 페이지의 『외부화 QMF 오브젝트의 세부사항』을 참조하십시오.

고정 형식 레코드

대부분의 레코드는 가변 형식을 가집니다. 그러나 레코드를 포함하는 파일 또는 데이터 세트가 가변 형식으로 되어 있더라도 헤더 레코드는 고정 형식을 가집니다.

헤더 레코드(H)

이 레코드는 내보낸 서식, 보고서 또는 프롬프트 조회의 내용을 식별하는 데 사용됩니다. 헤더 레코드는 내보낸 파일의 첫번째 레코드입니다. 헤더 레코드는 오브젝트의 특성을 설명합니다.

헤더 레코드는 표11에서 설명된 정보를 포함합니다(별표는 필드를 가져오기 위해 필요함을 의미).

표 11. 헤더 레코드 정보

바이트 위치	정보 및 유형
01*	헤더 레코드 표시기(H)
02	공백
03-05*	제품 ID(QMF)
06	공백

QMF 오브젝트 가져오기 및 내보내기

표 11. 헤더 레코드 정보 (계속)

바이트 위치	정보 및 유형
07-08	서식, 보고서 또는 프롬프트 조회를 내보낸 QMF 릴리스 레벨: QMF 버전 7의 경우 11
09	공백
10*	오브젝트 유형: 서식의 경우 F, 보고서의 경우 R, 관계형 프롬프트 조회의 경우 T, ER 프롬프트 조회의 경우 E
11	공백
12-13*	QMF 오브젝트 레벨: 보고서의 경우 01, 서식의 경우 04, 프롬프트 조회의 경우 01(관계형 또는 ER)
14	공백
15*	오브젝트의 형식: 서식, 보고서 및 프롬프트 조회(관계형 또는 ER) 오브젝트를 내보내는 데 사용되는 형식의 경우 E
16	공백
17	오브젝트의 상태: E - 오류 포함(서식의 경우에만), W - 경고 포함, V - 유효
18	공백
19	전체 또는 부분적인 오브젝트 표시기: 전체 오브젝트의 경우 W
20	공백
21	오브젝트를 내보냈을 때 사용중인 자국어: 영어의 경우 E
22	공백
23*	가져온 경우 임시 기억영역에서 오브젝트에 대한 조치(대체 오브젝트의 경우 R)
24	공백
25-26	다음 각 레코드의 시작에서 제어 영역의 길이: 서식의 경우 01, 보고서의 경우 02, 프롬프트 조회의 경우 01(관계형 또는 ER)
27	공백
28-29	V 및 T 레코드에 지정된 정수 길이 필드의 길이(03)
30	공백
31-38	날짜소인: yy/mm/dd
39	공백
40-44	시간소인: hh:mm
45	공백
46-50	OS/2 [®] 오브젝트의 경우 SSSSS
51	공백
52-56	OS/2 오브젝트의 경우 DDDDD

H 레코드에서 오브젝트 레벨은 오브젝트의 형식이 변경되었음을 나타냅니다. 모든 오브젝트 형식은 오브젝트 레벨 01로 시작합니다. QMF의 후기 릴리스가 오브젝트 형식을 변경했다면 오브젝트 레벨은 1씩 증가됩니다. 형식의 변경이 잠재적으로 응용프로그램에 오류를 야기하는 경우에만 오브젝트 레벨이 증가합니다. 94 페이지의 그림16의 코드화 형식을 가진 보고서, 서식 및 프롬프트 조회의 오브젝트 유형의 레벨 변경을 검사하십시오.

예를 들어 서식 오브젝트의 외부 형식은 버전 3.2에서 이전 릴리스와는 달리 구분 정보를 처리합니다. 이 변경 때문에 서식 오브젝트에 대한 오브젝트 레벨은 버전 3.2의 경우 03에서 04로 증가되었습니다. 일반적으로 다음 변경은 오브젝트 레벨이 증가되는 원인이 됩니다.

- V 또는 R 레코드에서 필드 번호가 삭제되거나 대체됩니다.
- 특정 레코드 유형의 레이아웃이 재정의됩니다.

그러나 필드에 대한 새 값 또는 새 필드 번호는 응용프로그램에 오류를 야기하지 않습니다. 가져오는 오브젝트가 응용프로그램에 문제를 일으키지 않는지 오브젝트 레벨 값을 검사하십시오.

프롬프트 조회용 H 레코드 예제

H QMF 11 T 01 E V W E R 01 03 98/11/20 17:12

예제의 값	설명
H QMF 11 T	버전 7 QMF 관계형 프롬프트 조회 헤더 레코드
01	프롬프트 조회의 구조는 오브젝트 레벨 1입니다.
E	형식 유형은 서식, 보고서 및 프롬프트 조회에 대한 것입니다.
V	내보낸 프롬프트 조회에는 어떠한 오류나 경고도 없습니다.
W	파일은 전체 프롬프트 조회를 포함합니다.
E	오브젝트를 내보낼 때 사용된 자국어는 영어입니다.
R	가져올 때 임시 기억영역 오브젝트가 대체됩니다.
01	제어 영역의 길이는 1바이트입니다.
03	정수 길이 필드의 길이는 3바이트입니다.
98/11/20	날짜소인
17:12	시간소인

QMF 오브젝트 가져오기 및 내보내기

프롬프트 조회 코드화 형식에 대한 예제를 보려면 115 페이지의 그림18을 참조하십시오.

서식용 H 레코드 예제

H QMF 11 F 04 E V W E R 01 03 98/12/16 22:08

예제의 값	설명
H QMF 11 F	버전 7 QMF 서식 헤더 레코드
04	서식의 구조는 오브젝트 레벨 4입니다.
E	형식 유형은 서식, 보고서 및 프롬프트 조회에 대한 것입니다.
V	내보낸 서식에는 어떠한 오류나 경고도 없습니다.
W	파일은 전체 서식을 포함합니다.
E	오브젝트를 내보낼 때 사용된 자국어는 영어입니다.
R	가져올 때 임시 기억영역 오브젝트가 대체됩니다.
01	제어 영역의 길이는 1바이트입니다.
03	정수 길이 필드의 길이는 3바이트입니다.
98/12/16	날짜소인
22:08	시간소인

서식 코드화 형식에 대한 자세한 예제를 보려면 119 페이지의 그림19를 참조하십시오.

보고서용 H 레코드 예제

H QMF 11 R 01 E V W E R 02 03 98/10/14 16:20

예제의 값	설명
H QMF 11 R	버전 7 QMF 보고서 헤더 레코드
01	보고서의 구조는 오브젝트 레벨 1입니다.
E	형식 유형은 서식, 보고서 및 프롬프트 조회에 대한 것입니다.
V	내보낸 보고서에는 어떠한 오류나 경고도 없습니다.
W	파일은 전체 보고서를 포함합니다.
E	오브젝트를 내보낼 때 사용된 자국어는 영어입니다.
R	무시됨
02	제어 영역의 길이는 2바이트입니다.
03	정수 길이 필드의 길이는 3바이트입니다.

예제의 값	설명
98/10/14	날짜소인
16:20	시간소인

보고서 코드화 형식에 대한 예제를 보려면 129 페이지의 그림21을 참조하십시오.

가변 형식 레코드

고정 형식 레코드인 H 레코드를 제외한 모든 레코드는 가변 형식 레코드입니다.

표시기 레코드 유형

- V** 데이터 값(104 페이지의 『데이터 값 레코드(V)』 참조)
- T** 데이터 표 설명(105 페이지의 『데이터 표 설명 레코드(T)』 참조)
- R** 표 행(109 페이지의 『표 행 레코드(R)』 참조)
- E** 오브젝트의 종료(111 페이지의 『오브젝트의 종료 레코드(E)』 참조)
- *** 응용프로그램 데이터(111 페이지의 『응용프로그램 데이터 레코드(*)』 참조)
- L** 보고서 행(130 페이지의 『보고서 행 레코드(L)』 참조)
- C** 데이터 연속(132 페이지의 『데이터 연속 레코드(C)』 참조)

가변 형식 레코드는 입력이 허용됩니다. 이것은 레코드를 포함하는 파일, 데이터 세트 또는 CICS 데이터 대기열이 아닌 레코드 자체를 나타냅니다. 가변 형식 레코드의 일반적인 서식은 다음과 같습니다.

제어 영역	레코드 데이터 영역
-------	------------

제어 영역은 다음과 같습니다.

바이트 위치

설명

- 01** 레코드 ID(H,V,T,R,E,*,L,C)
- 02** 공백(경우에 따라 생략됨. 가변 형식 레코드의 특정 유형을 참조하십시오).

레코드 데이터 영역은 특정 레코드에 관한 정보를 포함하는 가변 길이 영역입니다. 이 영역의 필드는 구분문자에 의해 분리됩니다(이 책에서는 공백 문자가 사용됩니다).

QMF 오브젝트 가져오기 및 내보내기

데이터 값 레코드(V)

값 레코드는 서식에서 표제 앞의 공백 행처럼 오브젝트의 단일 필드에 값을 제공하는 데 사용됩니다. V 레코드는 다음을 포함합니다.

- 오브젝트에 고유한 필드 번호
- 필드의 길이
- 필드의 값

251 페이지의 『부록B. 내보내기가져오기 형식』은 프롬프트 조회, 서식 및 보고서 오브젝트에 포함된 필드로 필드 번호를 할당합니다.

V 레코드의 내용은 다음과 같습니다.

V 레코드의 제어 영역

바이트 위치 설명

- | | |
|----|--|
| 01 | 값 레코드 ID(V) |
| 02 | 공백(보고서의 경우에만 사용되고, 서식 및 프롬프트 조회의 경우에는 생략됨) |

V 레코드의 레코드 데이터 영역

바이트 위치 설명

- | | |
|-------|--|
| 01 | 공백 |
| 02-05 | 필드 번호(1001-9999) |
| 06 | 공백 |
| 07-09 | 데이터 값의 길이(000-999). 별표(*) 다음에는 두 개의 공백이 올 수 있습니다. 별표는 데이터 값이 레코드의 종료에 의해 구분됨을 표시합니다. |
| 10 | 공백 |
| 11-종료 | 데이터 |

주:

1. 레코드 데이터 영역 바이트 위치는 제어 영역의 종료로부터 오프셋됩니다. 길이는 헤더 레코드에 표시됩니다.
2. 생략된 데이터 값(레코드의 종료 또는 길이 필드 다음에 오는 공백에서만)은 필드가 널 값을 포함함을 나타냅니다.

3. 길이 필드가 0이면 필드에 대해 기본값이 적용되고 경고 메시지를 발행합니다.
4. 지정된 길이가 다음에 오는 실제 데이터와 다르면 QMF는 경고를 발행합니다.

V 레코드 예제

서식: V 1511 * NONE

(필드 번호의 자세한 목록은 256페이지를 참조하십시오.)

필드 자동 준비된 보고서 행의 너비

값 'NONE'

보고서: V 1001 006 PERIOD

(필드 번호의 자세한 목록은 261페이지를 참조하십시오.)

필드 프로파일 DECIMAL 옵션

길이 6

값 'PERIOD'

프롬프트 조화: V 1501 001 K

(필드 번호의 자세한 목록은 254페이지를 참조하십시오.)

필드 중복 행

길이 1

값 유지

데이터 표 설명 레코드(T)

코드화된 형식에서, 오브젝트의 대부분의 데이터는 표에 나타납니다. 이는 데이터 베이스의 관계형 표가 아니라 코드화 형식 내의 그룹화 정보의 의미입니다.

서식에서 요약 계산처럼 각 T 레코드는 하나의 표를 정의하고 각 표는 오브젝트의 특정 부분과 일치합니다. 따라서 하나의 내보낸 파일은 여러 코드화된 표를 포

QMF 오브젝트 가져오기 및 내보내기

함합니다. 코드화된 표 및 열의 필드 번호에 대한 내용은 251 페이지의 『부록B. 내보내기/가져오기 형식』을 참조하십시오.

T 레코드 다음에는 항상 R 레코드가 옵니다. T 레코드는 다음에 오는 R 레코드에 대해 설명합니다. T 레코드 다음에 R 레코드가 오지 않으면 표는 생략됩니다.

응용프로그램이 R 레코드에 값을 제대로 위치시키려면 T 레코드에서 코드화를 사용하여 내보낸 서식, 보고서 또는 프롬프트 조회의 표 내용을 참조해야 합니다. 응용프로그램은 R 레코드의 정보를 찾기 위해 고정된 오프셋을 사용해서는 안됩니다.

T 레코드의 내용은 다음과 같습니다.

T 레코드의 제어 영역:

바이트 위치	설명
01	표 레코드 ID(T)
02	공백(보고서의 경우에만 사용되고, 서식 및 프롬프트 조회의 경우에는 생략됨)

T 레코드의 레코드 데이터 영역

다음 목록에서 바이트 위치는 제어 영역의 종료 다음에 오프셋되며 길이는 헤더 레코드에 표시됩니다.

바이트 위치	설명
01	공백
02-05	표 번호(1001-9999)
06	공백
07-09	이 표에서 행(R 레코드)의 수. 숫자 값 대신에 사용되는 별표(*)는 표가 다음에 오는 모든 R 레코드로 구성되어 있음을 의미합니다.
10	공백
11-13	레코드의 열 수(000-999)
14	공백

15-18, 24-27, ...

이 열의 필드 번호(반복 필드)

19, 28, ...

공백(반복 필드)

20-22, 29-31, ...

이 열의 데이터 값 길이(반복 필드)

바이트 11-13(열의 수)은 다음에 오는 필드 번호와 데이터 값 길이의 쌍이 몇 개인지 나타냅니다. 즉 15에서 22바이트까지에서 정보가 각 열마다 반복됩니다.

T 레코드 예제(서식)

T 1110 * 002 1112 007 1113 018

(필드 번호의 자세한 목록은 256페이지를 참조하십시오.)

필드	열 표제 표
행	모두
열	2
열 필드	열 데이터 유형
길이	7
열 필드	열 표제
길이	18

T 레코드 예제(프롬프트 조회)

T 1110 008 002 1112 001 1113 027

(필드 번호의 자세한 목록은 254페이지를 참조하십시오.)

필드	표 정의 표
행	8
열	2
열 필드	표 ID
길이	1
열 필드	표 이름
길이	27

R 레코드 예제(보고서)

T 1010 005 003 1012 008 1013 003 1014 006

(필드 번호의 자세한 목록은 261페이지를 참조하십시오.)

필드	형식이 지정된 보고서 표
행	5
열	3
열 필드	BREAKn
길이	8
열 필드	데이터의 편집 코드
길이	3
열 필드	데이터를 포함한 필드의 시작 위치
길이	6

규칙 및 주의사항:

1. 서식 또는 프롬프트 조회를 가졌로 때 R 레코드의 수는 T 레코드의 레코드 데이터 영역의 바이트 07-09에 지정된 행 계수와 일치해야 합니다. 그렇지 않으면, QMF는 경고를 발행합니다.
2. 서식 또는 프롬프트 조회를 가져올 때 바이트 11-13에 표시된 열의 수는 다음에 오는 바이트의 필드 번호/길이 쌍과 일치해야 합니다. 그렇지 않으면 QMF는 경고를 발행합니다.
3. 필드 번호와 길이 쌍은 표의 열 수에 의해 제한되고 순서는 임의적입니다.
4. 임시 기억영역에서 오브젝트가 갱신되고 경고가 발행하면 길이가 0인 열(또는 이 표에 포함되지 않은)은 기본값으로 설정됩니다. 그러나 프롬프트 조회의 경우 항상 그렇지 않습니다. 가능한 경우에 기본값이 제공됩니다. 그렇지 않으면, 오류가 발생합니다.
5. 열 필드를 공백으로 설정하려면 열은 T 레코드에 양수 길이를, R 레코드에는 공백 값을 가져야 합니다.

표 행 레코드(R)

R 레코드는 코드화된 표의 한 행에 값 세트를 제공합니다. R 레코드는 연관된 T 레코드에서 설명하는 순서대로 배열된 값 목록을 포함합니다. R 레코드는 T 레코드에 지정된 데이터 값의 위치와 길이에 대한 설명과 일치합니다. R 레코드의 내용은 다음과 같습니다.

R 레코드의 제어 영역

바이트 위치

설명

01 행 레코드 ID(R)

02 공백(보고서의 경우에만 사용되고 서식 및 프롬프트 조회의 경우에는 생략됨)

R 레코드의 레코드 데이터 영역

제어 영역 다음에 오는 R 레코드 데이터 영역은 구분문자(공백 문자)에 의해 분리된 일련의 값으로 구성되어 있습니다. 형식은 다음과 같습니다.

_val..._val..._val..

QMF 오브젝트 가져오기 및 내보내기

여기서 val...은 이 행 및 열의 데이터 값이고 _는 구분문자입니다.

R 레코드 예제

이 예제에서 열 값의 길이는 항상 해당 열의 T 레코드에 제공됩니다.

서식: R 2 SALARY

(필드 번호의 자세한 목록은 256페이지를 참조하십시오.)

열 값 ' 2'

열 값 'sALARY'

보고서: R GROUP L2

(필드 번호의 자세한 목록은 261페이지를 참조하십시오.)

열 값 'GROUP '

열 값 'L2'

프롬프트 조희: R C A.DEPT

(필드 번호의 자세한 목록은 254페이지를 참조하십시오.)

열 값 'C'

열 값 'A.DEPT'

규칙 및 주의사항:

1. R 레코드는 다른 R 또는 T 레코드 바로 다음에 와야 합니다.
2. 데이터 값의 수(v..v)는 연관된 T 레코드에서의 설명과 일치해야 합니다.
3. 연관된 T 레코드에서 데이터 값 길이가 0이면 오브젝트의 이 행과 열에 어떠한 값도 적용되지 않음을 나타냅니다. 즉, 이 값은 기본값으로 설정됩니다. 그러나 T 레코드에 필드가 있으면 R 레코드에는 이 필드에 대한 추가 공백이 있어야 합니다(길이 값이 0이면 R 레코드에서 하나의 공백 다음에 다른 것이 옵니다).

오브젝트의 종료 레코드(E)

E 레코드는 내보낸 오브젝트의 종료를 지정합니다. 이 레코드는 내보낸 파일의 마지막 레코드로서 문자 E로 나타납니다. 내보낸 보고서의 경우, E 레코드 다음에는 제어 영역을 완료하는 공백 문자가 오게 됩니다. 서식의 경우 공백은 생략됩니다.

E 레코드 다음에 오는 모든 레코드는 무시됩니다. E 레코드가 가져오는 중인 파일 내에 포함되어 있지 않으면 QMF는 파일의 종료값이 오브젝트의 종료를 나타내는 것으로 가정합니다.

응용프로그램 데이터 레코드(*)

응용프로그램 데이터 레코드는 응용프로그램이 주석과 같이 외부 파일에 제공된 오브젝트와 연관된 자신의 데이터를 포함하도록 허용합니다. 응용프로그램은 종종 파일에서 오브젝트를 자세히 설명하는 주석 레코드로서 이 레코드를 사용합니다. 별표 다음에 오는 정보는 기본적으로 무시되어 입력 프로세스에는 영향을 미치지 않습니다.

응용프로그램 데이터 레코드는 외부 파일에서 헤더(H) 레코드 앞을 제외한 어떤 위치에도 나타날 수 있습니다. QMF는 내보내기시 응용프로그램 데이터(*) 레코드를 작성하지 않습니다. 그러나, 작성한 파일, 데이터 세트 또는 CICS 데이터 대기열에서 이 레코드를 사용할 수 있습니다. 이 레코드는 주석 레코드로서 유용합니다. 응용프로그램 데이터 레코드의 내용은 다음과 같습니다.

바이트 위치 설명

01 응용프로그램 데이터 레코드 ID(*)

02-레코드의 종료
 데이터

응용프로그램 데이터 레코드 예제

*이것은 DEPT별로 그룹화되는 서식입니다.

이 주석은 내보낸 서식에 존재할 수 있습니다.

코드화 형식 오브젝트 내보내기

코드화 형식으로 오브젝트를 내보낼 경우:

- 모든 표 및 필드 번호는 네 자리수로 작성됩니다.
- 표 열은 오브젝트에 일반적으로 나타나는 순서에 따라 작성되며, 최대 길이를 가진 열이 표 레코드와 연관된 행 레코드의 오른쪽 끝으로 이동된다는 점만 다릅니다.
- 숫자의 길이는 세 자리수이고 필요하다면 맨 앞 0을 포함합니다.
- 공백 문자는 모든 레코드에서 구분문자로서 사용됩니다.
- 구분문자는 각 레코드의 마지막 문자 다음에 작성되지 않습니다.
- 공백은 모든 예약된 필드에 작성됩니다.
- E 레코드는 출력 파일로 작성된 마지막 레코드입니다.

코드화 형식 오브젝트 가져오기

서식, 보고서 또는 프롬프트 조화를 가져올 경우:

- 파일은 가변 또는 고정 길이 레코드로 구성되어 있습니다. 137 페이지의 『외부화 QMF 오브젝트의 세부사항』 및 251 페이지의 『부록B. 내보내기가져오기 형식』을 참조하십시오.
- 레코드 ID(H, V, T, R, E, *, L 또는 C)는 모든 레코드의 첫번째 위치에 존재해야 합니다.
- 첫번째 두 바이트는 제어 정보용으로 예약됩니다(제어 영역).
- 모든 데이터 필드(필드 번호, 길이 및 값 포함) 앞뒤에는 하나의 구분문자가 옵니다. 예외: 레코드에 있는 마지막 데이터 필드 다음에는 구분문자가 올 필요가

없는데, 이유는 레코드의 끝이 구분문자처럼 기능하기 때문입니다(이 책의 예제에서는 구분문자로 공백 문자를 사용했습니다).

- QMF가 IMPORT중에 중복 데이터 값 또는 표를 발견하면 이전 값 또는 표를 대체합니다. 그러나 특정 오브젝트에 대한 규칙을 지키지 않는 위치에서는 중복이 허용되지 않습니다. 예를 들어, 서식에 대해 제공된 열의 수는 첫번째 COLUMNS 표가 처리된 후에 변경할 수 없습니다.
- 표 번호, 필드 번호, 숫자의 길이는 맨 앞 0 또는 맨 앞 공백을 포함할 수 있습니다. 그러나 맨 뒤 공백(공백 구분문자의 경우 예외)은 허용되지 않습니다. 필드는 오른쪽에 맞추어야 합니다.
- *가 길 또는 계수 대신 사용되면 왼쪽에 맞추어 지고 맨 뒤 공백으로 채워집니다.
- 데이터 항목 필드에 제공된 값이 필드보다 짧은 경우 맨 뒤 공백으로 채워집니다. 값이 더 길면 잘립니다.
- 레코드가 고정 형식 길이보다 짧으면 지정되지 않은 상태의 필드는 공백으로 지정됩니다.

프롬프트 조회 오브젝트

이 절에서는 관계형 프롬프트 조회의 외부 형식에 대해 다룹니다.

내보낸 프롬프트 조회 오브젝트는 프롬프트 조회 1차 패널의 응답 영역에 표시되는 정보를 포함합니다. 내보낸 프롬프트 조회 파일, 데이터 세트 또는 CICS 데이터 대기열은 QMF 임시 기억영역 또는 데이터베이스로 가져올 수 있습니다. 프롬프트 조회를 가져오면 QMF는 입력 조회가 데이터베이스의 데이터와 일관되는지 검사합니다. 예를 들어, 가져올 프롬프트 조회 표 XYZ에 열 A, B 및 C가 있으면 QMF는 열 A, B 및 C를 가진 표 XYZ가 데이터베이스에 존재하는지 확인합니다.

프롬프트 조회 오브젝트와 연관된 표 및 필드 번호의 목록에 대해서는 254 페이지의 『프롬프트 조회 오브젝트의 표 및 필드 번호』를 참조하십시오.

프롬프트 조회 오브젝트 내보내기

이 절에서는 내보내는 프롬프트 조회의 예제에 대해 설명합니다. 그림17은 내보낸 프롬프트 조회의 프롬프트 조회 기본 패널 응답 텍스트입니다.

```
Tables:
  Q.STAFF(A)
  Q.ORG(B)
  Q.STAFF(C)

Join Tables:
  A.DEPT And B.DEPTNUMB
  And A.ID And C.ID

Columns:
  A.ID
  A.DEPT
  A.JOB
  A.SALARY
  DEPTNUMB
  C.SALARY
  C.SALARY+A.COMM

Row Conditions:
  If A.SALARY Is Greater Than 10000
  And A.DEPT Is Equal To 84 or 96

Sort:
  Descending by C.SALARY+A.COMM

Duplicate Rows:
  Keep duplicate rows
```

그림 17. 내보낸 프롬프트 조회 샘플

115 페이지의 그림18에서는 내보낸 프롬프트 조회의 형식을 보여줍니다.

```

H QMF 11 T 01 E V W E R 01 03 98/11/20 17:12
T 1110 003 002 1112 001 1113 050
R A Q.STAFF
R B Q.ORG
R C Q.STAFF
T 1150 002 002 1152 020 1153 020
R A.DEPT          B.DEPTNUMB
R A.ID            C.ID
T 1210 007 002 1212 001 1213 255
R C A.ID
R C A.DEPT
R C A.JOB
R C A.SALARY
R C B.DEPTNUMB
R C C.SALARY
R C C.SALARY+A.COMM
T 1310 009 003 1312 001 1313 008 1314 255
R 1 C            A.SALARY
R 2 IS          GT
R 3              10000
R 4 I
R 1 C            A.DEPT
R 2 IS          EQ
R 3              84
R 3              96
R 4 A
T 1410 001 002 1412 001 1413 255
R D C.SALARY+A.COMM
V 1501 001 K
E
    
```

그림 18. 내보낸 파일, 데이터 세트 또는 CICS 데이터 대기열

프롬프트 조회 가져오기

응용프로그램이 편집하거나 생성한 프롬프트 조회 오브젝트를 가져오려면 다음을 알아야 합니다.

- 프롬프트 조회 파일을 가져오려면 입력 레코드는 헤더(H) 레코드 다음에 특정 순서로 되어 있어야 합니다. 순서는 다음과 같습니다.

1. 표 정의의 T 레코드
2. 표 이름의 R 레코드

QMF 오브젝트 가져오기 및 내보내기

3. 열 정의의 T 레코드

4. 열의 R 레코드

5. 행 조건 레코드(필드 번호 1310)는 항목 유형 순서 번호(필드 번호 1312)에 따라 각 조건 내에서 순서대로 정렬됩니다. 즉, 행 데이터가 프롬프트 조회 응답 영역에 나타나는 것과 같은 순서입니다.

6. 나머지 레코드는 어떤 순서로 와도 좋습니다.

- Tables 표는 다른 표 또는 V 레코드 앞에 나타나야 합니다.
- Tables T 레코드에서 행 계수의 값은 * 또는 0에서 15까지의 정수가 됩니다. 행 계수 값이 0이면 조회의 모든 내용이 무시되는데, 이는 비어 있는 조회가 가져오기됨을 의미합니다.
- QMF는 프롬프트 조회 가져오기에 대해 경고를 발행하지 않습니다.
- 두 번째 Tables 표(표 1110)가 지정되면 QMF는 오류를 발행하고 표의 내용은 무시됩니다.
- 프롬프트 조회는 가져오기시 기본값을 제공하지 않습니다.
- 정렬 표가 있으면 그 앞에 나오는 열 표가 있습니다.
- QMF는 가져오기 파일에서 중복 레코드를 허용합니다. 레코드의 경우 가장 최근의 값이 사용됩니다.
- 모든 열 이름은 가져오기 동안 표 ID에 의해 규정화됩니다.
- 프롬프트 조회를 사전 할당된 데이터 세트로 내보내면 허용되는 최소의 논리 레코드 길이(LRECL)는 259바이트입니다.
- 프롬프트 조회의 내보내기 형식은 사용되는 자국어에 관계없이 동일합니다. 형식은 언어에 따라 다릅니다. 헤더 레코드에서 언어 바이트는 가져오기중에는 무시됩니다. 254 페이지의 『프롬프트 조회 오브젝트의 표 및 필드 번호』에서 프롬프트 조회를 내보낼 때 사용되는 코드를 볼 수 있습니다.

요약 함수 및 표현식은 변환되지 않습니다. 그러므로 요약 함수 COUNT, AVG, SUM, MIN 및 MAX는 변경되지 않은 채로 남아 있습니다. 이들은 변환되지 않는 SQL 기호입니다.

서식 오브젝트

서식 오브젝트는 모든 QMF 서식 패널에서 지정된 모든 정보를 포함합니다. 서식을 내보내면 QMF는 사용자가 변경한 서식 패널을 코드화 형식으로 변환합니다. 다음 패널은 사용자가 패널을 변경한 경우에만 코드화 형식으로 존재합니다.

- FORM.BREAK n , 여기서 n 은 3에서 6까지임
- FORM.CALC
- FORM.CONDITIONS
- FORM.DETAIL에 대해 1보다 큰 모든 변이 패널

외부화 형식에서 사용되지 않은 패널을 삭제하면 시스템의 공간이 절약됩니다.

기본 서식 작성: 예제

LAYOUT 명령은 QMF 임시 기억영역에서 존재하는 데이터 없이도 서식(QMF 임시 기억영역에서 또는 데이터베이스에서)에 기반하여 샘플 보고서를 사용자가 볼 수 있도록 합니다. LAYOUT은 샘플 데이터를 생성하고, 그것을 QMF로 가져오고, 서식을 적용하여 보고서를 작성합니다.

CICS 사용자를 위한 주의사항

LAYOUT 명령은 ISPF를 필요로 하는데, 이는 CICS에서는 사용할 수 없습니다.

사용자는 서식을 작성하고, 내보낸 다음, 이를 초기화 과정의 부분으로 QMF에 가져와서 조회를 실행하지 않고 서식을 볼 수 있습니다. 초기 프로시저중에 서식을 가져올 경우 사용자는 SHOW FORM을 입력함으로써 서식을 액세스할 수 있습니다.

가져올 수 있는 최소 서식은 헤더 및 종료 레코드입니다. 그러나 FORM.COLUMNS를 사용하려면 하나 이상의 정보 열이 있어야 합니다.

비어 있는 보고서를 작성하는 조회를 실행함으로써 기본 서식을 작성할 수 있습니다.

QMF 오브젝트 가져오기 및 내보내기

SQL 조회

```
SELECT JOB  
FROM Q.STAFF  
WHERE NAME='empty_set'
```

QMF가 보고서를 표시할 경우 EXPORT FORM TO DEFAULT를 입력하십시오(CICS에서 (QUEUETYPE=xx 매개변수 포함). DEFAULT로 명명된 파일, 데이터 세트 또는 CICS 데이터 대기열은 119 페이지의 그림19에 표시된 정보를 포함합니다.

H QMF 11 F 04 E V W E R 01 03 98/12/16 22:08

```

T 1110 001 011 1112 007 1113 040 1114 007 1115 006 1116 005 1117 005 1118 003 1119 008 1120 008
  1122 006 1121 050
R CHAR      JOB                2      5      C      1  DEFAULT
  DEFAULT NO
V 1201 001 0
V 1202 001 2
T 1210 001 003 1212 004 1213 006 1214 055
R 1        CENTER
V 1301 001 2
V 1302 001 0
T 1310 001 003 1312 004 1313 006 1314 055
R 1        CENTER
V 1401 002 NO
V 1402 001 1
V 1403 001 0
T 1410 001 003 1412 004 1413 006 1414 055
R 1        RIGHT
V 1501 001 1
V 1502 003 YES
V 1503 003 YES
V 1504 003 YES
V 1505 003 YES
V 1506 003 YES
V 1507 003 YES
V 1508 003 YES
V 1509 003 YES
V 1510 003 YES
V 1511 004 NONE
V 1512 002 NO
V 1513 007 DEFAULT
V 1514 002 NO
V 1515 004 NONE
V 2790 001 1
V 2791 003 YES
V 2805 003 YES
T 2810 001 003 2812 004 2813 006 2814 055
R 1        LEFT
V 2901 002 NO
V 2902 001 1
V 2904 001 0
V 2906 002 NO
V 2907 002 NO
T 2910 001 003 2912 004 2913 006 2914 055
R 1        LEFT
V 3080 001 1
V 3101 002 NO
V 3102 002 NO
V 3103 001 0
V 3104 001 0
T 3110 001 003 3112 004 3113 006 3114 055

```

그림 19. 내보낸 서식의 샘플 형식 (1/2)

QMF 오브젝트 가져오기 및 내보내기

```
R 1    LEFT
V 3201 002 NO
V 3202 001 1
V 3203 001 0
V 3204 001 1T 3210 001 003 3212 004 3213 006 3214 055
R 1    RIGHT
V 3080 001 2
V 3101 002 NO
V 3102 002 NO
V 3103 001 0
V 3104 001 0
T 3110 001 003 3112 004 3113 006 3114 055
R 1    LEFT
V 3201 002 NO
V 3202 001 1
V 3203 001 0
V 3204 001 1
T 3210 001 003 3212 004 3213 006 3214 055
R 1    RIGHT
E
```

그림 19. 내보낸 서식의 샘플 형식 (2/2)

초기 프로시저에서 `IMPORT FORM FROM DEFAULT`(CICS에서 (`QUEUETYPE=xx` 매개변수 포함) 명령을 발행함으로써 사용자가 로그인할 때마다 기본 파일, 데이터 세트 또는 CICS 데이터 대기열을 가져올 수 있습니다.

응용프로그램에서 QMF 서식 오브젝트에 대한 고려사항

응용프로그램에서 QMF 서식을 사용하려면 다음 사항을 숙지해야 합니다.

- QMF 외부에서 서식 파일, 데이터 세트 또는 CICS 데이터 대기열을 작성할 경우

QMF 외부에서 서식 `&file`을 작성하려면(즉 `EXPORT FORM`을 사용하여 작성하지 않음), 그것을 QMF에 성공적으로 가져오기 위해 완전한 서식 오브젝트가 필요하지 않습니다. 정말 필요한 것은 `COLUMNS` 표의 T 및 R 레코드 앞에 오는 헤더(H) 레코드입니다. 그것을 가져오면 서식의 나머지는 기본값이 적용됩니다.

사용자 고유의 서식 파일, 데이터 세트 또는 CICS 데이터 대기열을 작성할 경우 보다 융통성이 있어야 합니다. `EXPORT FORM`을 사용한 경우에는 확보한 파일, 데이터 세트 또는 CICS 데이터 대기열과 똑같지는 않아도 됩니다. 예

를 들어 QMF가 서식을 내보내면 값(V) 레코드의 모든 데이터 값 앞에는 길이가 옵니다. 이에 반하여 서식을 가져올 때에는 데이터 값이 레코드의 종료에 의해 구분됨을 지정하는 별표(*)를 사용할 수 있습니다.

가져온 서식에서 R 레코드 계수가 기본 서식의 연관된 영역에 QMF가 이미 할당한 기본 행의 수보다 적으면 QMF는 초과 행을 그대로 유지합니다.

- 헤더 레코드에서 오브젝트 레벨을 검사할 경우

서식 파일, 데이터 세트 또는 CICS 데이터 대기열의 헤더 레코드의 오브젝트 레벨은 서식이 생성되었을 때의 형식 구조 레벨을 알려줍니다(오브젝트 레벨은 99페이지에서 설명된 것처럼 헤더 레코드의 바이트 12 및 13에서 표시됩니다). 오브젝트 레벨이 응용프로그램이 기반한 형식을 나타내는지 검사하여 응용프로그램이 서식 파일, 데이터 세트 또는 CICS 데이터 대기열의 내용을 올바르게 해석하는지 확인할 수 있습니다.

- 응용프로그램 데이터 레코드를 사용할 경우

111 페이지의 『응용프로그램 데이터 레코드(*)』에서 언급된 응용프로그램 데이터 레코드는 응용프로그램에서 유용합니다. 이들은 서식 오브젝트의 파일, 데이터 세트 또는 CICS 데이터 대기열에 자체 주석을 추가할 수 있게 해 줍니다. 주석을 헤더 레코드 다음에 오는 파일, 데이터 세트 또는 CICS 데이터 대기열 내에 어디에나 배치할 수 있습니다. QMF가 그러한 레코드를 읽으면, * 다음에 오는 레코드의 모든 데이터가 무시됩니다. 따라서 레코드는 가져오기 프로세스에 어떠한 영향도 주지 않습니다.

- 날짜/시간 정보를 가져오고 내보낼 경우

설치가 날짜/시간 데이터 유형을 지원하고 날짜/시간 정보가 있는 서식을 내보낼 경우 그 이후에 날짜/시간 데이터 유형을 지원하지 않는 QMF 설치를 사용하는 서식을 가져올 수 없습니다. 그렇게 하면, IMPORT 명령 처리가 구분되고 QMF는 오류 메시지를 발행합니다.

- QMF 버전 3.2에서 변경된 구분 필드 번호

QMF 버전 3.2는 각 구분 패널에 표 및 필드 번호를 포함시키지 않고 어떤 구분 패널이 그 다음에 오는 정보를 수신할지 나타내기 위해 『트리거』로서 작동하는 하나의 필드 번호(3080)를 현재 사용하고 있습니다.

QMF 오브젝트 가져오기 및 내보내기

내보낸 파일, 데이터 세트 또는 CICS 데이터 대기열에서 구분 패널을 작성할 경우 필드 3080을 원하는 구분 패널의 수로 설정할 수 있습니다. 이 필드의 유효한 값은 1부터 6까지만입니다.

모든 패널을 정의하지 않고 코드화된 파일, 데이터 세트 또는 CICS 데이터 대기열에서 6개의 구분 패널 중 하나를 사용할 수 있습니다. 예를 들어, 구분 패널 1에서 4까지를 작성하지 않고도 Break5 패널을 작성할 수 있습니다.

- 서식 응용프로그램 마이그레이션 지원

QMF가 버전 3.2에서 구분 패널에 대한 새 필드 번호를 도입하였으며 이 새 번호가 QMF의 초기 버전과 호환될 수 없기 때문에 내보낸 서식의 오브젝트 레벨은 오브젝트 레벨 4까지 증가합니다. 서식 오브젝트는 QMF 버전 3 릴리스 1과 버전 3 릴리스 1 수정판 1(버전 3.1.1) 사이에서나 버전 3.1.1과 버전 3 릴리스 2(버전 3.2) 사이에서는 변경되지 않았습니다.

QMF 버전 2.4 또는 이전 버전에서(QMF VSE 버전 1 포함) QMF 버전 3.2까지 업그레이드할 경우 새 구분 필드 번호를 반영하도록 현재 응용프로그램을 업그레이드해야 합니다. 그러나 QMF 버전 3.2는 버전 3.2를 설치할 수 있도록 지원하는 서식 응용프로그램 마이그레이션 지원을 제공하고 이전 구분 필드 번호를 사용하는 기존의 응용프로그램을 여전히 사용합니다.

QMF 버전 3.2 서식 오브젝트를 내보내면 이 지원은 서식 오브젝트의 새 구분 필드 번호를 QMF 버전 2.4 사본으로 변환시킵니다. 기존 응용프로그램은 즉시 갱신하지 않고도 QMF 버전 3.2로 실행할 수 있습니다.

중요: 서식 응용프로그램 마이그레이션 지원은 QMF 버전 3.2의 내보내기 및 QMF 버전 2.4와 함께 사용하는 것을 허용하지 않습니다.

서식 응용프로그램 마이그레이션 지원을 설정하는 데 대한 내용을 보려면 *Installing and Managing QMF for VM/ESA* 또는 *Installing and Managing QMF for MVS*를 참조하십시오.

- CICS에서 서식 사용시 제한사항

REXX는 QMF CICS에서 사용할 수 없기 때문에 CICS 환경에서 서식을 실행하려 하면 REXX에 따라 달라지는 QMF의 영역은 작동하지 않습니다. 이 영역은 FORM.CALC 패널, FORM.CONDITIONS 패널 및 정의 지정 창에 입력된 모든 내용을 포함합니다. 따라서 REXX 계산, 조건부 행 형식화 및 열 정의는 QMF CICS 사용자의 경우 사용할 수 없습니다.

서식 파일, 데이터 세트 또는 CICS 데이터 대기열의 입력 및 출력에 대한 자세한 내용과 규칙은 112 페이지의 『코드화 형식 오브젝트 가져오기』를 참조하십시오.

서식 오브젝트 가져오기

서식을 가져올 때 이 필드에는 대문자로 입력해야 합니다.

- 모든 레코드의 레코드 ID
- 헤더 레코드에는 다음 필드가 있습니다.
 - 제품 ID(QMF)
 - 오브젝트 유형(F)
 - 오브젝트 형식(E)
 - 오브젝트에 대한 조치(R)
- COLUMNS 표의 R 레코드의 데이터 유형 값(NUMERIC, CHAR, GRAPHIC, UNKNOWN). 설치가 낱짜/시간 데이터 유형을 지원하면 데이터 유형 값 DATE, TIME 및 TIMEST도 역시 대문자여야 합니다.
- 모든 서식 키워드 및 대체 변수가 서식 패널에서 사용됩니다. 서식을 가져오면 서식의 모든 입력이 그대로 남아 있습니다. 서식 키워드가 소문자이면 서식 패널에서 오류 표시기가 켜집니다. 오류를 정정하려면 필드 위에 겹쳐서 입력해야 합니다. 데이터 유형 값이 대문자로 입력되지 않으면 오류가 발생하고 IMPORT는 종료됩니다.

COLUMNS 표의 T 레코드(필드 번호 1110)는 헤더 레코드 바로 다음에 와야 하며, 여기에는 표 행의 수에 대한 숫자 계수가 포함되어야 합니다(* 행 계수는 허용되지 않음).

전체 COLUMNS 표를 읽으면 지정되지 않은 필드는 기본값으로 설정되며 서식이 표시됩니다.

변이 패널

변이 번호 필드(필드 번호 2790)는 필드 뒤에 나오는 모든 변이 패널 정보에 의해 갱신될 변이 패널을 판별합니다. 이 V 레코드는 주어진 변이 패널의 모든 다른 V, T 및 R 레코드보다 먼저 나옵니다.

QMF 오브젝트 가져오기 및 내보내기

특정 변수에 대한 값이 코드화된 형식에 두 번 이상 나타나면 최근 값은 원래 값으로 대체됩니다. 서식에서 변수의 수는 서식의 최상위 변수 번호와 같습니다. 가져올 때 변수 번호에는 순서가 필요없습니다.

변환된 서식

영어 서식을 비영어 세션으로 가져올 경우 H 레코드에서 자국어 ID가 E이면 QMF는 자동으로 서식의 예약어를 현재 세션의 언어로 변환합니다.

가져온 서식에서 데이터 유형, 편집 코드 및 너비 생략

COLUMNS 표에서 다음 규칙이 준수되면 데이터 유형(필드 번호 1112), 편집 코드(필드 번호 1117) 및 너비(필드 번호 1116)는 선택적으로 생략됩니다.

- 데이터 유형 및 너비가 생략되면 편집 코드가 포함되어야 합니다. 지정된 편집 코드에 따라, QMF는 편집 코드 및 너비에 대해 적절한 기본값을 삽입합니다.
- 편집 코드 및 너비가 생략되면 데이터 유형이 포함되어야 합니다. QMF는 편집 코드 및 너비에 대해 기본값을 제공합니다.
- 너비는 데이터 유형 또는나 편집 코드 중 하나와 함께 사용되어야 합니다.

표12는 열 데이터 유형 필드의 값에 대한 정보를 포함합니다.

표 12. 열 데이터 유형 필드의 값

데이터 유형 값 (QMF 서식)	데이터 유형 번호 (데이터베이스 서식)	문자열(데이터베이스 서식)	의미
DATE	384	DATE	날짜
TIME	388	TIME	시간
TIMEST	392	TIMESTAMP	시간소인
	496	INTEGER	정수
	500	SMALLINT	작은 정수
	484	DECIMAL	십진수
NUMERIC	480	FLOAT	부동 소수점
			가변 문자
	448	VARCHAR	고정 문자
	452	CHAR	긴 가변
	456	LONG VARCHAR	문자
CHAR	904	ROWID	행 ID

표 12. 열 데이터 유형 필드의 값 (계속)

데이터 유형 값 (QMF 서식)	데이터 유형 번호 (데이터베이스 서식)	문자열(데이터베이스 서식)	의미
			가변 그래픽
	464	VARGRAPHIC	고정 그래픽
	468	GRAPHIC	긴 가변
GRAPHIC	472	LONG VARGRAPHIC	그래픽

실행하는 데이터 유형 값과 함께 QMF가 U, V 또는 유효하지 않은 편집 코드에 응답할 때 사용하는 UNKNOWN 데이터 유형이 있습니다.

가져오기 중 오류 검출

가져오는 동안 QMF가 서식 파일의 형식에서 오류를 발견하면 가져오기 기능은 오류를 설명하는 메시지 및 파일 내의 위치와 함께 종료됩니다.

헤더 레코드에 오류가 발생하고 서식이 임시 기억영역에 이미 존재하면 기존의 서식이 표시됩니다. 서식을 성공적으로 가져오면 QMF는 서식 패널을 표시합니다.

헤더 레코드를 읽은 후 오류가 발생되면 임시 기억영역에 있는 기존의 모든 서식은 폐기되고 시작 패널이 표시됩니다. 그러나 데이터 오브젝트가 존재하면 QMF는 데이터에 대한 기본 서식을 생성하지만, 표시하지는 않습니다.

QMF에 의해 발견되는 어떠한 작은 오류라도 가져오기를 종료하지는 않습니다. 그러한 경우 QMF는 경고 메시지를 발행하고 적절한 위치에 기본값을 적용합니다. 예를 들면 다음과 같습니다.

V 레코드

- 길이가 0인 필드
- 지정된 길이 필드는 실제로 제공된 데이터의 길이와 일치하지 않습니다.

T 레코드

- 길이가 0인 열
- 지정된 열의 수가 다음 필드 번호/길이 쌍과 일치하지 않습니다.

다음 두 방법 중 하나로 오류 및 경고에 응답할 수 있습니다.

- 동시에 한 문제점을 수정합니다.
- SET PROFILE로 IMPORT FORM 명령을 실행합니다(TRACE=L2).

QMF 오브젝트 가져오기 및 내보내기

SET PROFILE(TRACE=L2)과 함께 IMPORT FORM 명령을 실행하면 사용자에게 IMPORT 명령과 관련된 모든 메시지 번호 목록이 제공됩니다. 명령은 다음과 같습니다.

```
HELP message_number
```

해당 메시지 번호에 대한 오류 메시지를 표시합니다.

서식 오브젝트 내보내기

QMF 버전 1 릴리스 2 전에 내보낸 서식의 형식은 이 절에 설명된 형식과 다르지만, QMF는 계속해서 초기 릴리스의 QMF를 통해 내보내는 서식을 허용합니다. 그러한 서식이 최근 릴리스에 가져오거나 표시되면 성능 저하를 가져올 수도 있습니다.

256 페이지의 『서식 오브젝트의 표 및 필드 번호』는 서식 오브젝트의 다양한 부분에 할당된 필드 번호를 나열합니다. 또한 서식의 어떤 부분이 표이고 어떤 부분이 내보낸 파일의 개별 값인지 표시합니다. 열 데이터 유형(필드 1112)은 서식 패널에는 표시되지 않지만, 외부 형식을 가진 서식과 연관됩니다.

서식을 가져온 경우 열 데이터 유형은 필요하지 않습니다. 가져오는 동안 이 열 데이터 유형이 소실되면 QMF는 편집 코드를 통해 기본 데이터 유형 정보를 제공합니다. 자세한 내용은 123 페이지의 『서식 오브젝트 가져오기』를 참조하십시오. 내보내는 동안 QMF가 제공하는 데이터 유형 키워드(필드 번호 1112)는 지정된 편집 코드에 기반합니다. U, V 또는 유효하지 않은 편집 코드의 경우 QMF는 데이터 유형 키워드 UNKNOWN을 지정합니다. 127 페이지의 표13에서는 지정된 편집 코드에 대해 QMF가 생성한 데이터 유형 키워드를 보여줍니다. 이 표에서 x는 표시할 소수 자리수를 나타냅니다. 여기서 x는 0 - 99 사이의 정수입니다.

표 13. 지정된 편집 코드에 대해 생성된 데이터 유형 키워드

지정된 편집 코드	데이터 유형 키워드
B, BW, C, CW, CT, CDx, X, XW	CHAR
G, GW	GRAPHIC
E, D, I, J, K, L, P, EZ, DZ, IZ, JZ, KZ, LZ, PZ, DZC, Dx, Ix, Jx, Kx, Lx, Px	NUMERIC
TDXx	DATE
TTXx	TIME
TSI	TIMEST
U, V	UNKNOWN
위의 것 중 해당되는 것이 없습니다(유효하지 않음).	UNKNOWN

변이 패널

서식을 내보내면 QMF는 기본값에서 변경된 값을 가진 변이 패널만 내보냅니다. 그러므로 외부 서식에서 변이의 총 수는 패널에 있는 변이 계수 표시기에 표시된 수보다 적을 수도 있습니다. QMF는 변이 번호를 변경하여 변이를 연속되는 순서로 다시 정렬할 수 있습니다.

변환된 서식

비 영어 세션에서 서식을 내보내면 현재 세션 언어 또는 영어 중 하나로 서식을 내보낼 수 있습니다. 따라서 H 레코드의 자국어 ID는 서식을 내보내는 세션의 언어를 반영하지 않을 수도 있습니다.

보고서 오브젝트

QMF가 보고서를 표시하면 임시 기억영역에서 서식과 데이터 오브젝트간의 대화의 결과를 볼 수 있습니다. 보고서 오브젝트는 임시 기억영역에 존재하지 않습니다. 보고서를 내보내면 QMF는 실제로 서식 및 데이터 오브젝트의 대화를 내보냅니다. 보고서는 데이터베이스에서 저장될 수 없고 내보낸 보고서를 QMF로 다시 가져올 수 없습니다. 그러나 내보낸 보고서를 다음과 같이 사용할 수 있습니다.

- 보고서에서 데이터를 추출하고 응용프로그램에서 사용합니다.
- 응용프로그램에 따라 인쇄하거나 화면에 다시 표시하기 위해 보고서 모양을 수정합니다.

보고서 내보내기 예제

이 예제는 레벨 1 구분을 포함하는 보고서에 대해 설명합니다. 그림20에서는 내보낼 보고서를 보여줍니다. 가로 보고서의 예제에 대해서는 133페이지를 참조하십시오.

필드 번호의 목록을 보려면 261 페이지의 『보고서 오브젝트의 표와 필드 번호』를 참조하십시오.

보고서		행 1	POS 1	79
J & H SUPPLY COMPANY AVERAGE SALARIES (DEPTS 10, 15, 20) REPORT 17				
DEPT	JOB	AVERAGE SALARY		
-----	-----	-----		
10	MGR	20865.86		
	*	20865.86		
15	CLERK	12383.35		
	MGR	20659.80		
	SALES	16502.83		
	*	15482.33		
20	CLERK	13878.68		
	MGR	18357.50		
	SALES	18171.25		
	*	16071.53		
		=====		
		17473.24		
COMPANY NAME REPORT 17				

그림 20. 표 형식 QMF 보고서

129 페이지의 그림21에서는 그림20에 표시된 내보낸 보고서의 형식을 보여줍니다.

```

H QMF 11 R 01 E V W E R 02 03 98/10/14 11:24
V 1001 006 PERIOD
V 1002 003 016
T 1010 003 006 1013 005 1014 006 1015 006 1016 006 1017 006 1012 008
R L 000001 000003 000008 000001 BREAK1
R C 000009 000011 000015 000001 GROUP
R L2 000016 000018 000027 000001 AVERAGE
L 110 10100000
L 110 10100000
L 110 10100000
L 110 10000000
L 110 10000000
L 170 10000000
L 170 10000000
L 170 10010000
L 181 11000000
L 151 10010000
L 151 11100000
L 151 10000000
L 181 11000000
L 181 11000000
L 181 11000000
L 151 10010000
L 151 11100000
L 151 10000000
L 181 11000000
L 181 11000000
L 181 11000000
L 151 10010000
L 151 11100000
L 151 10000000
L 190 10010000
L 190 11000000
L 120 10000000
L 120 10000000
L 120 10100000
L 120 10100000
E

```

J & H SUPPLY COMPANY
AVERAGE SALARIES (DEPTS 10, 15, 20)
REPORT 17

	DEPT	JOB	AVERAGE
			SALARY
L 170 10000000	-----	-----	-----
L 181 11000000	10	MGR	20865.86
L 151 10010000			-----
L 151 11100000		*	20865.86
L 151 10000000			
L 181 11000000	15	CLERK	12383.35
L 181 11000000		MGR	20659.80
L 181 11000000		SALES	16502.83
L 151 10010000			-----
L 151 11100000		*	15482.33
L 151 10000000			
L 181 11000000	20	CLERK	13878.67
L 181 11000000		MGR	18357.50
L 181 11000000		SALES	18171.25
L 151 10010000			-----
L 151 11100000		*	16071.52
L 151 10000000			
L 190 10010000			=====
L 190 11000000			17473.24
L 120 10000000			
L 120 10000000			
L 120 10100000			
L 120 10100000			

COMPANY NAME
보고서 17

그림 21. 내보낸 샘플 보고서의 형식

보고서를 내보내는 경우, QMF는 보고서의 내용을 해석하기 위해 추가 정보로 형식이 지정된 보고서의 전체 텍스트를 작성합니다.

QMF 오브젝트 가져오기 및 내보내기

헤더 레코드는 내보낸 파일의 첫번째 레코드입니다. 그것은 해당 V, T 및 R 레코드의 앞에 옵니다. 보고서가 가로 유형 보고서이면 첫번째 그룹 다음에 다른 그룹인 V, T 및 R 레코드가 나옵니다.

내보낸 보고서 역시 H, V, T, R 및 E 레코드와 함께 두 개의 추가 레코드의 유형을 필요로 합니다.

보고서 행 레코드(L)

데이터 연속 레코드(C)

이 두 레코드는 V, T 및 R 레코드의 마지막 그룹 다음에 옵니다.

응용프로그램에서 형식이 지정된 보고서 데이터만 사용하려면 QMF가 인쇄 출력을 파일, 데이터 세트 또는 CICS 데이터 대기열로 보내게 하십시오. 이 파일, 데이터 세트 또는 CICS 데이터 대기열은 어떠한 레이아웃 정보없이도 형식이 지정된 데이터만 포함합니다. 파일, 데이터 세트 또는 CICS 데이터 대기열을 인쇄하도록 프린터를 설정하는 데 대한 내용은 다음을 참조하십시오.

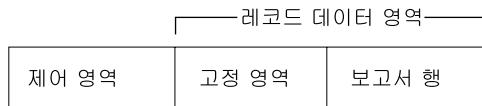
Installing and Managing QMF for MVS,

Installing and Managing QMF for VM/ESA 또는

Installing and Managing QMF for VSE/ESA

보고서 행 레코드(L)

보고서에서 형식이 지정된 각 행은 L 레코드에 의해 설명됩니다. 보고서에는 각 행마다 하나의 L 레코드가 있습니다. 다른 가변 형식 레코드(V, T, R)와 같이 L 레코드도 레코드 데이터 영역의 앞에 오는 제어 영역으로 구성되어 있습니다. 제어 영역의 형식은 다른 레코드와 유사합니다. 레코드 데이터 영역은 형식이 지정된 보고서 행 앞에 나오는 고정 영역으로 구성되어 있습니다. 고정 영역은 다음에 오는 보고서 행에 관한 정보를 제공합니다.



L 레코드의 내용은 다음과 같습니다.

L 레코드의 제어 영역

바이트 위치 설명

- 01 값 레코드 ID(L)
- 02 연속 표시기. 현재 레코드가 데이터 연속 레코드로 계속되는 여부를 표시합니다(132 페이지의 『데이터 연속 레코드(C)』 참조).
 - 연속의 경우 C
 - 현재 레코드의 끝과 다음 레코드 데이터 부분의 시작에 삽입된 DBCS 구분문자 S0와 SI로 연속된 경우 D
 - 연속되지 않으면 공백

(관련 설명을 보려면 132 페이지의 2 및 1을 참조하십시오.)

L 레코드의 레코드 데이터 영역(고정 영역):

바이트 위치	설명
01	공백
02-04	보고서 부분 표시기 110 = 페이지 표제 120 = 페이지 각주 13n = 구분 표제(n은 구분 번호, 1-6) 15n = 구분 각주(n은 구분 번호, 1-6) 170 = 열 표제 171 = 세부 표제 180 = 세부 행 181 = 그룹 요약 행 190 = 마지막 각주
05	공백
06-13	행 유형 속성. 바이트 06은 항상 1입니다. 바이트 7에서 13까지에서 각 바이트는 형식화된 보고서 행에서 일치하는 행 유형 속성이 있는지 여부를 표시합니다(1 = 속성 있음, 0 = 속성 없음).
바이트 위치	설명
06	1
07	데이터
08	텍스트
09	분리자
10	열 자동 줄바꿈. 주 3을 참조하십시오.
11	행 자동 줄바꿈. 주 3을 참조하십시오.
12	두 번째 데이터 행(가로 보고서 전용). 주 4를 참조하십시오.
13	예약됨
14	공백

id=repline.L 레코드의 레코드 데이터 영역(보고서 행):

바이트 위치	설명
--------	----

QMF 오브젝트 가져오기 및 내보내기

01-종료 실제 형식화된 보고서 행

L 레코드 예제

```
L 151 11100000 DEPARTMENT TOTALS 93,659.45
```

(텍스트와 데이터를 포함하는 구분1 각주 행)

주:

1. C 레코드는 제어 영역의 바이트 2에 연속 문자로 표시된 L 레코드 다음에 옵니다.
2. D가 제어 영역의 연속 표시기에 대해 지정되면 이것은 현재 레코드가 너무 길어 하나의 실제 레코드에 맞출 수 없음을 의미합니다. 또한 레코드의 분할이 진행되는 중에 SO(shift out) 및 SI(shift in) 문자가 현재 또는 다음 레코드에 추가되어 계속되는 DBCS 데이터의 무결성을 보존합니다.
3. 열 자동 줄바꿈(바이트 10) 및 행 자동 줄바꿈(바이트 11)에 대한 속성은 단일 논리 보고서 행이 복수의 실제 보고서 행에 연속됨을 나타내는 데 사용됩니다. 주어진 L 형식 레코드에 속성이 존재할 경우 열 데이터 또는 자동 줄바꿈된 행이 다음 L 형식 레코드로 이어짐을 의미합니다.
4. 퍼센트 또는 누계 열을 포함하는 가로 보고서는 각 그룹(또한 구분 및 마지막) 요약별로 두 개의 데이터 행을 포함할 수 있습니다. 첫번째 요약 데이터 행은 고유한 각 가로 값을 가로로 계산한 누적 퍼센트 또는 열의 누계 값을 포함합니다. 두 번째 요약 데이터 행은 각 그룹을 세로로 계산한 열의 누적 퍼센트 또는 누계 값을 포함합니다 (보고서나 제어 구분 내에서). 두 번째 데이터 행(바이트 12)의 행 유형은 이러한 성향을 가진 내보낸 보고서 또는 두 번째 데이터 행을 식별합니다.

데이터 연속 레코드(C)

둘 이상의 레코드에 값 또는 값 세트를 연속으로 포함시키는 데 C 레코드가 사용됩니다. 이러한 연속 형태는 계속되는 레코드 바로 다음에 옵니다. C 레코드의 형식은 계속되고 있는 원래 레코드의 형식과 일치합니다. QMF는 C 레코드를 사용하여 L 레코드만을 계속합니다. C 레코드는 다음을 포함합니다.

C 레코드의 제어 영역

바이트 위치 설명

01 값 레코드 ID(C)

- 02** 연속 표시기. 현재 레코드가 다른 C 레코드로 계속되는지 표시합니다.
- 연속인 경우 C
 - 현재 레코드의 끝과 다음 레코드 데이터 부분의 시작에 삽입된 DBCS 구분문자 S0와 SI로 연속된 경우 D
 - 연속되지 않으면 공백
- (132 페이지의 1 및 132 페이지의 2를 참조하십시오.)

C 레코드의 레코드 데이터 영역

다음 목록에서 바이트 위치는 제어 영역의 끝으로부터의 오프셋이고 길이는 헤더 레코드에 표시됩니다.

바이트 위치	설명
01	공백
02-끝	계속되는 값 또는 값 세트

C 레코드 예제

단일 보고서 행 값이 텍스트 중간에서 분리되지만 다른 연속 레코드로 계속되지 않는 보고서 행 연속인 경우:

```
C  ARS ----> <----- TOTAL ---->
```

단일 보고서 행 값이 텍스트 중간에서 분리되고 레코드는 다른 C 레코드로 계속되는 보고서 행 연속인 경우:

```
CC ERK ----> <----- MGR ----> <----- SAL
```

가로 보고서의 내보내기 예제

134 페이지의 그림22는 내보낸 가로 유형 보고서에 대해 설명합니다.

QMF 오브젝트 가져오기 및 내보내기

보고서 행 1 POS 1 79

J & H SUPPLY COMPANY
DEPT AVERAGE SALARIES
보고서 18 (가로 보고서)

DEPT	<----- JOB -----> <- CLERK --> AVERAGE SALARY	<-- MGR ---> AVERAGE SALARY	<- SALES --> AVERAGE SALARY	<- TOTAL --> AVERAGE SALARY
10		20865.86		20865.86
15	12383.35	20659.80	16502.83	15482.33
20	13878.68	18357.50	18171.25	16071.53
38	12482.25	17506.75	17407.15	15457.11
	=====	=====	=====	=====
	12914.76	19998.21	17372.10	16880.26

회사 이름

보고서 18
페이지 1

그림 22. 샘플 가로 보고서. 이 보고서는 QMF 가로 보고서 기능을 사용합니다.

135 페이지의 그림 23에서는 그림 22로부터 생성된 결과 코드화 형식을 보여줍니다.

H QMF 11 R 01 E V W E R 02 03 98/10/14 16:20

```

V 1001 006 PERIOD
V 1002 003 016
T 1010 002 006 1013 005 1014 006 1015 006 1016 006 1017 006 1012 008
R L      000001 000003 000008 000001 GROUP
R L2     000003 000005 000014 000001 AVERAGE
V 2001 005 C
V 2002 003 001
V 2003 003 YES
T 2010 004 003 2012 006 2013 006 2014 006
R 000014 000018 000009
R 000029 000031 000023
R 000042 000046 000037
R 000056 000060 000051
L 110 10100000
L 110 10100000
L 110 10100000
L 110 10000000
L 110 10000000
L 170 10000000
L 170 11000000
L 170 10000000
L 170 10000000
L 170 10010000
L 181 11000000
L 181 11000000
L 181 11000000
L 181 11000000
L 190 10010000
L 190 11000000
L 120 10000000
L 120 10000000
L 120 10100000
L 120 10100000
L 120 10100000
E
    
```

J & H SUPPLY COMPANY
DEPT AVERAGE SALARIES
보고서 18 (가로 보고서)

	<----- JOB ----->				
	<- CLERK -->	<- MGR ---->	<- SALES -->	<- TOTAL -->	
	AVERAGE	AVERAGE	AVERAGE	AVERAGE	
DEPT	SALARY	SALARY	SALARY	SALARY	
-----	-----	-----	-----	-----	
10		20865.86		20865.86	
15	12383.35	20659.80	16502.83	15482.33	
20	13878.68	18357.50	18171.25	16071.53	
38	12482.25	17506.75	17407.15	15457.11	
	=====	=====	=====	=====	
190	12914.76	19998.21	17372.10	16880.26	

회사 이름
보고서 18
페이지 1

그림 23. 내보낸 가로 유형 보고서의 형식

HTML 보고서

HTML 보고서를 내보낼 경우 QMF가 필요한 HTML 태그를 보고서의 앞뒤에 배치시키면 사용자는 웹 서버에 배치할 수 있고 HTML 3.0이 적용되는 웹 브라우저에 표시할 수 있습니다. 136 페이지의 그림24에서는 QMF가 보고서에 배치하는 HTML 코딩을 보여줍니다.

QMF 오브젝트 가져오기 및 내보내기

```
<HTML>
<HEAD>
<TITLE>
보고서
</TITLE>
</HEAD>
<BODY>
<PRE>

                J & H SUPPLY COMPANY
      AVERAGE SALARY (DEPTS 10, 15, 20)
                보고서 17

      DEPT  JOB          AVERAGE
      ----  -          -
      10    MGR          20865.86
                        *  20865.86

      15    CLERK        12383.53
           MGR          20659.80
           SALES        16052.83
                        *  15482.33

      20    CLERK        13878.67
           MGR          18357.50
           SALES        18171.25
                        *  16071.52
                        =====
                        17473.52

                        회사 이름
                        보고서 17

</PRE>
</BODY>
</HTML>
```

그림 24. 샘플 HTML 보고서 코딩

QBE 조회

QBE 조회 오브젝트를 QMF 내부 형식을 사용하여 내보냅니다. 이 형식은 어떠한 방법으로도 변경할 수 없습니다.

외부화 QMF 오브젝트의 세부사항

표14는 TSO와 CMS IMPORT 및 EXPORT 파일의 세부사항을 포함합니다.

CICS의 경우, 레코드 크기는 표14에 나타난 것과 같지만 반드시 그래야 하는 것은 아닙니다. 예를 들어, 레코드 크기가 32K인 임시 기억영역 대기열로부터 SQL 조회를 가져올 수 있고 QMF는 이를 79바이트로 잘라냅니다.

레코드 형식은 CICS 임시 기억영역 또는 임시 데이터 대기열에 대한 요소가 아닙니다. 임시 기억영역 대기열은 형식에 관계없이 레코드를 보유하고 있습니다. 임시 데이터 대기열은 목적지 제어 표(DCT)로 레코드 형식을 무시합니다.

대기열 이름은 사용자가 작성한 것이며 기본 접두어 또는 접미어가 없습니다. CICS TS 대기열 이름은 8바이트입니다. TD 대기열 이름은 4바이트입니다.

표 14. 파일 및 데이터 세트 속성

오브젝트	레코드 크기	레코드 형식(CMS/TSO)
데이터 또는 표 (QMF 형식)	최대 크기: 7000바이트	레코드는 고정 길이여야 합니다.
데이터 또는 표(IXF 형식)	최대 크기 : 32756(주 2 참조). 가져오기 동안 QMF가 IXF 파일, 데이터 세트 또는 CICS 데이터 대기열에 허용하는 최소 LRECL는 49바이트입니다.	레코드는 가변 길이여야 합니다.
프롬프트 조회	최대: 7290바이트 최소: EXPORT시 266바이트. IMPORT시 41바이트	레코드는 EXPORT시 가변 길이여야 합니다. IMPORT시에는 고정 길이 또는 가변 길이 중 하나가 될 수 있습니다.
SQL 조회	EXPORT시 79바이트여야 합니다. IMPORT시 256바이트보다 적어야 하지만 잘려져서 79바이트가 됩니다.	레코드는 EXPORT시 고정 길이여야 합니다. IMPORT시에는 고정 길이 또는 가변 길이 중 하나가 될 수 있습니다.
QBE 조회	1024바이트여야 합니다(주 3 참조).	레코드는 가변 길이여야 합니다.
서식	최대: 7290바이트 최소: EXPORT시 161바이트. IMPORT시 23바이트	레코드는 EXPORT시 가변 길이여야 합니다. IMPORT시에는 고정 길이 또는 가변 길이 중 하나가 될 수 있습니다.
Proc	EXPORT시 79바이트여야 합니다. IMPORT시 어떤 크기도 될 수 있지만, 잘려서 79바이트가 됩니다.	레코드는 EXPORT시 고정 길이여야 합니다. IMPORT시에는 고정 길이 또는 가변 길이 중 하나가 될 수 있습니다.

QMF 오브젝트 가져오기 및 내보내기

표 14. 파일 및 데이터 세트 속성 (계속)

오브젝트	레코드 크기	레코드 형식(CMS/TSO)
보고서	최대: 7290바이트 최소: 65바이트	레코드는 가변 길이여야 합니다.
HTML 보고서	최대: 32000바이트	레코드는 가변 길이여야 합니다.

주:

1. EXPORT 또는 IMPORT 명령에 파일, 데이터 세트 또는 CICS 데이터 대기열의 이름을 지정해야 합니다. 이름에 관한 자세한 내용은 *QMF* 참조서를 참조하십시오.
2. 정의된 열을 포함하는 내보낸 서식의 최소 LRECL은 161바이트입니다. 이 최소값은 열 표제 정렬, 열 데이터 정렬, 열 정의 표현식 및 널(null) 제공에 대한 정보를 포함하는 QMF 서식에 대한 버전 3.2 기능 강화를 수용합니다. 서식이 열 정의 정보를 포함하지 않으면 CMS의 최소 LRECL은 113바이트입니다.
3. 빈 QBE 조회는 828바이트입니다.
4. 레코드 크기를 일반적으로 내보내고 있는 표에서 데이터 행의 길이(널 표시기 및 DBCS 구분문자의 공간 포함)에 IXF D 유형 레코드 계수 필드(5바이트)의 길이를 합한 길이입니다. 행 길이로부터 얻어진 레코드 크기가 가장 긴 IXF 헤더 레코드의 길이(81바이트)보다 짧을 경우 레코드 크기는 81바이트로 설정됩니다.

CICS 대기열 사용 규칙 및 고려사항

규칙:

1. CICS에서 IMPORT와 EXPORT 모두에 대해 QUEUETYPE 옵션을 지정해야 합니다. 기본값은 없습니다.
2. CICS의 임시 데이터(TD) 대기열로부터 오브젝트를 가져올 경우 정확한 오브젝트 유형을 지정해야 합니다. QMF가 한 번 내용을 검색하면 대기열은 공백이 됩니다. 예를 들어, 임시 데이터 대기열의 오브젝트 유형이 프로시저일 경우 『서식』을 지정하면 QMF는 오류 메시지를 발행합니다. 그러나 해당 대기

열이 지금 비어 있기 때문에, 같은 대기열을 사용하여 IMPORT 명령을 다시 (올바른 오브젝트 유형을 사용하는 경우에도) 성공적으로 발행할 수 없습니다.

3. CICS에서, 임시 데이터 또는 임시 기억영역(TS) 대기열은 IMPORT 명령을 발행하기 전에 하나의 완전한 단일 QMF 오브젝트를 포함해야 합니다.
4. 임시 데이터 대기열을 내보낼 경우 EXPORT 명령을 발행하기 전에 대기열은 열린 상태여야 하고, 사용가능해야 하며, 비어 있어야 합니다. CICS 임시 데이터 대기열에 대한 내용은 *CICS for VSE/ESA Application Programming Guide*를 참조하십시오.

고려사항:

QMF는 CICS 임시 데이터 대기열을 임시 기억영역 대기열과 다르게 처리합니다.

- **임시 데이터 대기열:** QMF는 화면에 오브젝트를 표시하기 전에 전체 임시 데이터 대기열을 가져옵니다. 즉, 전체 대기열의 내용이 기억영역 또는 스펠 영역에 맞아야 합니다. 또한 표시될 오브젝트가 크면 QMF가 화면에 오브젝트를 표시하기 전에 지연이 발생할 수 있게 됩니다.

CICS 내부 파티션 임시 데이터 대기열은 최대 32K의 데이터 행을 보유할 수 있습니다. 외부 파티션 임시 데이터 대기열은 오브젝트를 보유할 수 있는 만큼 커질 수 있습니다.

- **임시 기억영역 대기열:** QMF에서는 사용자에게 표시하기 전에 약 100개 행의 임시 기억영역을 읽습니다. 임시 기억영역 대기열은 최대 32K의 데이터 행을 보유할 수 있습니다.

기억영역 고려사항 및 스펠 영역에 관한 내용은 *Installing and Managing QMF for VSE/ESA* 또는 *Installing and Managing QMF for MVS*를 참조하십시오.

- **대기열에 QMF 오브젝트 추가:** QMF는 IMPORT 및 EXPORT 명령에 SUSPEND 매개변수를 사용하여 CICS가 명령 실행 시기를 조정합니다.

IMPORT 및 EXPORT 명령에 있는 SUSPEND 매개변수는 대기열이 사용중일 경우 취해야 할 조치를 결정합니다. SUSPEND 매개변수가 YES로 설정되면 QMF는 CICS 데이터 대기열 이름에 대해 CICS ENQ(enqueue)를 발행합니다. 이것은 QMF 오브젝트를 대기열로 작성하기 전에 대기열을 사용할 수 있

QMF 오브젝트 가져오기 및 내보내기

을 때까지 기다리도록 CICS에 지시하므로 QMF 트랜잭션이 대기열에 의해 처리되는 작업을 방해하지 않게 해 줍니다.

SUSPEND 매개변수가 N0로 설정되면 EXPORT 명령은 취소되고 메시지가 리턴됩니다. SUSPEND 기본값은 N0입니다. QMF가 자동 ENQ를 발행하는 결과가 EXPORT 및 IMPORT 명령의 SUSPEND 옵션에 반영됩니다.

제9장 QMF 응용프로그램 디버깅

QMF는 오류 처리와 응용프로그램 지원 명령뿐 아니라 프로그램에 대한 디버깅 기능도 제공합니다. 이 장에서 설명하는 기술은 호출 인터페이스 응용프로그램에 적용됩니다.

ISPF 디버깅 기술에 관한 내용은 43 페이지의 『제5장 ISPF를 사용하는 QMF 응용프로그램 작성』을 참조하십시오. TRACE 문을 통해 REXX 추적 기능을 사용할 수 있습니다. 이 명령문에 대한 자세한 내용은 *REXX Reference*를 참조하십시오.

호출 인터페이스 응용프로그램 디버깅

QMF는 두 추적 옵션 L 및 A를 제공하고 몇 가지 추적 레벨의 응용프로그램 디버깅 기능을 제공합니다.

L 옵션을 사용한 추적

L 옵션을 사용하면 QMF 세션을 시작하기 전에 할당된 외부 QMF 추적 데이터 출력에 메시지 및 명령을 기록하도록 QMF에 지시할 수 있습니다. 다음과 같이 선택할 수 있는 두 가지 L 옵션이 있습니다.

- L1** 모든 QMF 메시지는 QMF 추적 데이터 출력으로 작성됩니다.
- L2** 모든 QMF 메시지 및 명령은 QMF 추적 데이터 출력으로 작성됩니다. 예를 들면, L2를 사용하여 Q.SYSTEM_INI 시스템 초기화 프로시저 명령 및 메시지를 추적 및 디버그할 수 있습니다.

다음 두 방법 중 하나로 L 옵션을 설정할 수 있습니다.

1. DISPLAY PROFILE 명령을 발행하고, PROFILE 오브젝트가 나타나면 L1 또는 L2 중 하나로 TRACE 옵션을 변경합니다.
2. 다음 명령을 발행하십시오.

```
SET PROFILE (TRACE=x)
```

여기서, x는 L1 또는 L2 중 하나입니다.

추적 데이터 출력을 직접 할당하면 인쇄되거나 터미널에서 표시될 추적 정보를 정렬할 수 있습니다. 어느 경우에도, QMF 세션 후에 데이터를 조사할 수 있습니다. 할당에 대한 자세한 내용은 144 페이지의 『QMF 추적 데이터 출력 할당』을 참조하거나 정보 센터에 문의하십시오.

A 옵션을 사용한 추적

A 옵션은 QMF 응용프로그램 지원 서비스의 추적 레벨을 지정할 수 있도록 합니다.

A 옵션 설정값은 A0, A1 또는 A2가 될 수 있습니다. A0는 기본값이고 A 추적 없음 신호로 해석됩니다. A1 및 A2는 보다 자세한 세부 결과를 보기 위해 호출할 수 있습니다. 이것은 다른 QMF 추적 옵션에 사용되는 패턴입니다. 이 추적 옵션 및 다른 추적 옵션에 관한 자세한 내용은 *Installing and Managing QMF for MVS*, *Installing and Managing QMF for VM/ESA* 또는 *Installing and Managing QMF for VSE/ESA*를 참조하십시오.

QMF SET 명령을 사용하거나 DISPLAY PROFILE 명령을 실행한 후에 화면에 입력하여 L 옵션을 지정했던 동일한 방법으로 A 옵션을 지정합니다. 예를 들면, 디버그중인 응용프로그램을 호출하기 바로 전에 다음을 입력할 수 있습니다.

```
SET PROFILE (TRACE=L2A1)
```

그 다음에 응용프로그램을 시작할 경우 L2 및 A1 추적이 모두 적용됩니다.

현재 A 옵션 설정 값을 판별하려면, DSQAO_APPL_TRACE 변수를 살펴보십시오. A0, A1 또는 A2 설정값은 각각 0, 1 또는 2입니다. DSQAO_APPL_TRACE 값을 사용하여 그림25에서와 마찬가지로 응용프로그램에서 원하는 추적의 종류를 선택할 수 있습니다.

```

/* REXX program to set tracing                               */
call dsqcix "GET GLOBAL(A_TRACE=DSQAO_APPL_TRACE)"
if a_trace > 0 then
do
  /* trace code for both A1 and A2 */
  :
  if a_trace = 2 then
  do
    /* trace code for just A2  */
    :
  end
end
end

```

그림 25. 추적을 설정하는 샘플 REXX 프로그램

그림 25의 경우처럼 중첩된 DO 그룹이 응용프로그램 전반에 나타날 수 있습니다. 이 그룹이 나타나는 곳에서 이 그룹은 특정 데이터 영역의 『스냅샷』 덤프를 구하거나, 어떤 중요한 변수의 값을 인쇄하거나, 디버깅 모듈을 로드하거나, 응용프로그램 디버깅을 도울 수 있는 다른 진단 프로시저를 실행합니다. 정확하게 수행되는 작업은 응용프로그램이 실행되는 동안 A 옵션에 대해 적용되는 설정값에 따라 달라집니다.

A 옵션 코드의 위치의 좋은 위치는 큰 응용프로그램 내부입니다. 디버깅을 종료한 후에 응용프로그램에 이 코드를 남겨두는 방법을 고려하십시오. A0 설정값으로 응용프로그램을 실행하면 A 추적 출력이 생성되지 않습니다. 응용프로그램을 수정했으나 프로세스에 버그가 있을 경우에는 이 코드를 다시 실행할 수 있습니다.

추적 해제

응용프로그램을 테스트한 후 추적을 해제하려면, 다음 명령을 발행하십시오.

```
SET PROFILE (TRACE=NONE)
```

이것은 QMF 세션의 나머지 부분에 대해 추적을 구분합니다. 그러나 이렇게 해도 영구 QMF 프로파일에는 영향을 주지 않습니다.

QMF 추적 데이터 출력 할당

추적이 사용될 경우 QMF를 호출하기 전에 QMF 추적 데이터 출력을 할당해야 합니다. 시동 프로시저를 통해 출력이 자동으로 할당되었을 수도 있습니다. 이렇게 해도, 원래의 할당이 사용자 요구를 충족시키지 못하면 출력을 다시 할당하려 할 수 있습니다.

CMS 또는 TSO의 QMF 추적 데이터 출력 할당 방법에 대한 예제는 해당 언어에 대한 장에서 어셈블 또는 컴파일 및 실행 코드 예제를 참조하십시오.

어셈블러

어셈블러 언어 인터페이스, 148페이지에서 시작

C 언어

C 언어 인터페이스, 171페이지에서 시작

COBOL

COBOL 언어 인터페이스, 189페이지에서 시작

FORTRAN

FORTRAN 언어 인터페이스, 207페이지에서 시작

PL/I PL/I 언어 인터페이스, 224페이지에서 시작

REXX

REXX 언어 인터페이스, 241페이지에서 시작

이 예제에서 명령은 QMF 세션이 종료한 후에 터미널에서 검사할 수 있는 순차적 추적 데이터 출력을 할당합니다. 출력은 고정 길이의 80자 레코드로 되어 있습니다. 추적 정보는 행당 80자로 형식화되어 있습니다. 터미널 화면에서 출력의 전체 행을 볼 수 있습니다.

CICS의 경우, 프로그램 매개변수 DSQSDBQT 및 DSQSDBQN을 사용하여 QMF가 추적 데이터를 놓는 위치를 지정할 수 있습니다. QMF는 많은 양의 추적 데이터를 생성할 수 있기 때문에 CICS가 임시 기억영역을 사용할 경우 주의해야 합니다. CICS 임시 기억영역은 메시지 또는 작은 응용프로그램 추적 데이터용으로만 사용하는 것이 좋습니다.

QMF 메시지 명령에 추적 사용

응용프로그램을 종료할 때 QMF MESSAGE 명령을 사용하면 메시지를 표시하는 것 이상의 작업을 수행할 수 있습니다. QMF 추적 데이터 출력에 메시지를 기록

할 때에도 이 명령을 사용할 수 있습니다. 이를 위해 TRACE의 L 옵션을 L1 또는 L2에 설정하여 응용프로그램을 실행하십시오(이 방법에 관한 내용은 141 페이지의 『L 옵션을 사용한 추적』을 참조하십시오). MESSAGE 명령을 사용하여 처리되는 모든 메시지는 다른 QMF 메시지(L2가 사용되는 경우에는 명령)와 함께 QMF 추적 데이터 출력에 기록됩니다.

프로그램의 전략적 위치에 MESSAGE 명령을 배치함으로써 QMF 추적 파일에 유용한 정보를 로그할 수 있습니다. QMF 세션 후에 터미널에서 또는 인쇄된 출력을 통해 이를 시험할 수 있습니다. QMF 추적 데이터 출력에 관한 자세한 내용은 144 페이지의 『QMF 추적 데이터 출력 할당』을 참조하십시오.

예제

응용프로그램에서는 다음 예제에 표시된 명령을 발행합니다.

```
call dsqcix "SET PROFILE (TRACE=L2"
:
:
call dsqcix "MESSAGE (TEXT='QUERYA COMPLETED SUCCESSFULLY'"
:
:
call dsqcix "MESSAGE (TEXT='EXECB ENTERED WITH VALUE OF 7'"
:
:
```

메시지 ‘QUERYA COMPLETED SUCCESSFULLY’ 및 ‘EXECB ENTERED WITH VALUE OF 7’을 포함하는 레코드는 QMF 추적 데이터 출력에 기록됩니다.

QMF 메시지는 릴리스에 따라 달라질 수 있기 때문에 QMF 추적 데이터 출력을 응용프로그램에 입력으로 사용해서는 안 됩니다.

START 및 다른 QMF 명령의 오류 디버깅

DSQCOMM의 레벨에 따라 DSQCOMM에 메시지 텍스트가 있을 수도 있습니다. START 명령(또는 임의의 QMF 명령)이 실패하면 이 메시지 텍스트는 디버깅할 필요가 없습니다. 현재 DSQCOMM의 레벨에서 작업할 경우 메시지 텍스트를 사용할 수 있습니다. 전체 QMF 오류 메시지에 관한 내용은 *QMF* 메시지 및 코드를 참조하십시오.

부록A. 호출 인터페이스 언어의 샘플 코드

이 부록에는 각 QMF 호출 인터페이스 언어의 샘플 코드가 있습니다.

어셈블러

148 페이지의 『어셈블러 언어 인터페이스』

C 언어

171 페이지의 『C 언어 인터페이스』

COBOL

189 페이지의 『COBOL 언어 인터페이스』

FORTRAN

207 페이지의 『FORTRAN 언어 인터페이스』

PL/I 224 페이지의 『PL/I 언어 인터페이스』

REXX

241 페이지의 『REXX 언어 인터페이스』

이 부록에는 QMF가 지원하는 각 언어에 대한 샘플 프로그램이 있습니다. 각 샘플 프로그램은 다음을 수행합니다.

- QMF를 시작합니다.
- 세 개의 글로벌 변수를 설정합니다.
- Q1이라는 조회를 실행합니다.
- 서식 F1을 사용하여 결과 보고서를 인쇄합니다.
- QMF 세션을 종료합니다.

QMF는 조회 Q1 또는 서식 F1을 제공하지 않으나 샘플 프로그램은 이 오브젝트를 작성된 그대로 사용합니다.

이 부록에서는 호출 인터페이스를 사용하여 프로그램을 어셈블하거나 컴파일하고, 링크 편집하고, 실행하는 방법을 보여줍니다. QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

어셈블러 언어 인터페이스

어셈블러 언어를 사용할 경우 호출 인터페이스와 함께 어셈블러 H(AH) 또는 고급 어셈블러(HLASM)를 사용해야 합니다. QMF는 어셈블러 언어를 위해 하나의 함수 호출 DSQCIA를 제공합니다.

CICS/VSE의 경우, 31비트 주소를 지정하려면 HLASM을 사용해야 합니다.

어셈블러에 대한 인터페이스 통신 영역 맵핑(DSQCOMMA)

DSQCOMMA는 어셈블러 언어에 DSQCOMM 맵핑을 제공하고 제품과 함께 제공됩니다. 표15에서는 DSQCOMMA 값을 보여줍니다.

표 15. DSQCOMMA의 인터페이스 통신 영역

구조 이름	데이터 유형	설명
DSQ_RETURN_CODE	DS F	QMF 명령이 실행된 후의 상태를 나타냅니다. 해당 값은 다음과 같습니다. DSQ_SUCCESS 요청의 성공적인 실행 DSQ_WARNING 경고 메시지를 나타내는 정상적인 완료 DSQ_FAILURE 명령이 제대로 실행되지 않음 DSQ_SEVERE 심각한 오류. QMF 세션이 종료됨
DSQ_INSTANCE_ID	DS F	START 명령의 실행 동안 QMF에 의해 설정된 ID
DSQ_COMM_LEVEL	DS CL12	DSQCOMM의 레벨을 식별합니다. 이것을 DSQ_CURRENT_COMM_LEVEL 값으로 설정한 다음 QMF START 명령을 발행합니다.
DSQ_PRODUCT	DS CL2	사용중인 IBM 조회 제품을 식별합니다.
DSQ_PRODUCT_RELEASE	DS CL2	사용중인 조회 제품의 릴리스 레벨을 식별합니다.
DSQ_RESERVE1	DS XL28	앞으로 사용되도록 예약됨
DSQ_MESSAGE_ID	DS CL8	완료 메시지 ID
DSQ_Q_MESSAGE_ID	DS CL8	조회 메시지 ID

표 15. DSQCOMMA의 인터페이스 통신 영역 (계속)

구조 이름	데이터 유형	설명
DSQ_START_PARM_ERROR	DS CL8	START가 매개변수 오류로 인해 실패했을 경우 오류 상태에 있는 매개변수
DSQ_CANCEL_IND	DS C	QMF 명령이 실행되는 동안 사용자가 취소했을 경우 다음 두 값 중 하나를 포함합니다. <ul style="list-style-type: none"> • DSQ_CANCEL_YES • DSQ_CANCEL_NO
DSQ_RESERVE2	DS XL23	앞으로 사용되도록 예약됨
DSQ_RESERVE3	DS XL156	앞으로 사용되도록 예약됨
DSQ_MESSAGE_TEXT	DS CL128	완료 메시지 텍스트
DSQ_Q_MESSAGE_TEXT	DS CL128	조회 메시지 텍스트

어셈블러 언어의 함수 호출

어셈블러 언어의 함수 호출에는 DSQCIA 및 DSQCIA 확장 구문의 두 가지 형식이 있습니다.

DSQCIA

이 호출은 응용프로그램 변수에 액세스를 요구하지 않는 QMF 명령에 대한 것입니다. 대부분의 QMF 명령에서 이 호출을 사용하십시오.

```
CALL DSQCIA, (DSQCOMM,CMDLTH,CMDSTR),VL
```

매개변수 값은 다음과 같습니다.

DSQCOMM

인터페이스 통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이, FULLWORD 매개변수

CMDSTR

실행할 QMF 명령, CMDLTH로 지정된 길이를 가진 대문자 문자열.

VL은 어셈블러 VARIABLE LIST문입니다.

DSQCIA, 확장 구문

DSQCIA 함수의 이 확장 구문 형식은 응용프로그램 변수에 액세스해야 하는 세 가지 QMF 명령인 START와 확장 형식 GET GLOBAL 및 SET GLOBAL에 대한 형식입니다.

```
CALL DSQCIA, (DSQCOMM,CMDLTH,CMDSTR,  
             PNUM,KLTH,KWORD,VLTH,VALUE,VTYP),VL
```

매개변수 값은 다음과 같습니다.

DSQCOMM

인터페이스 통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이, FULLWORD 매개변수

CMDSTR

실행할 QMF 명령, CMDLTH로 지정된 길이를 가진 대문자 문자열

PNUM

명령 키워드의 수, FULLWORD 매개변수

KLTH

각 지정된 키워드의 길이, FULLWORD 매개변수 또는 FULLWORD 매개변수의 배열

KWORD

QMF 키워드. 길이가 KLTH에 의해 지정된 것과 같은 문자 또는 문자 구조

VLTH

키워드와 연관된 각 값의 길이, FULLWORD 매개변수 또는 FULLWORD 매개변수의 배열

VALUE

각 키워드와 연관된 값. 유형은 VTYP 매개변수에 지정되어 있으며 문자, 문자의 구조, FULLWORD 매개변수 또는 FULLWORD 매개변수의 배열이 될 수 있습니다.

VTYP

값 문자열 VALUE의 QMF 데이터 유형. 이 유형은 통신 매크로 DSQCOMM에서 제공되는 두 값 중 하나입니다.

- 문자 값의 경우 DSQ_VARIABLE_CHAR. VTYPE이 DSQ_VARIABLE_CHAR이면, VALUE의 유효성이 검사되지 않습니다.
- 정수 값의 경우 DSQ_VARIABLE_FINT. VTYPE이 DSQ_VARIABLE_FINT이면, VALUE의 유효성이 검사되며 VALUE는 정수여야 합니다.

VALUE 필드에 지정된 모든 값은 VTYPE에 지정된 데이터 유형이어야 합니다.

VL은 어셈블러 VARIABLE LIST문입니다.

마이그레이션 정보

DSQCOMM은 버전 2 릴리스 4와 버전 3.2 사이에서 변경되었습니다.

- 이전 DSQCOMM를 계속해서 사용하려면 프로그램을 다시 어셈블할 필요가 없습니다.
- DSQCIA의 버전 3.2를 사용하려면 버전 2 릴리스 4 프로그램을 다시 링크 편집해야 합니다.

버전 3.2 DSQCOMM는 START 명령에 오류가 있는 경우 특히 유용한 메시지 텍스트를 제공합니다. 새 DSQCOMM을 사용하려면, 프로그램을 다시 어셈블한 후 DSQ_COMM_LEVEL(DSQCOMM에서)을 DSQ_CURRENT_COMM_LEVEL로 초기화해야 합니다. 이 값을 설정하지 않으면 QMF는 DSQCOMM을 버전 2 릴리스 4로 처리합니다.

버전 3.2 레벨의 DSQCOMM 길이는 512바이트이며 이는 버전 2 릴리스 4에서 사용가능한 256바이트로부터 증가되었습니다. 256바이트로 제한된(예: MVC) 이 구조를 이동시키거나 초기화하는 데 사용되는 명령은 좀더 광범위한 데이터 영역에서 작동하는 명령어를 사용할 수 있도록 변경되어야 합니다.

CICS/MVS 사용자를 위한 주의사항

DSQCIA는 버전 3 릴리스 1 수정판에서 버전 3 릴리스 2로 변경되었습니다. QMF 제공 함수 호출과 주 QMF 프로그램간의 인터페이스는 CALL 인터페이스에서 EXEC CICS LINK 인터페이스로 변경되었습니다. 새 인터페이스는 사용자 프로그램과 QMF 제품을 확실하게 구별해 줍니다. 인터페이스가 변경되었기 때문에 호출 인터페이스를 사용한 프로그램을 다시 링크 편집해야 합니다.

어셈블러 프로그래밍 예제

여기에서 샘플 소스 코드 목록을 볼 수 있고 온라인으로 액세스할 수도 있습니다.

- MVS의 경우, 샘플 프로그램은 라이브러리 QMF710.SDSQSAPE의 멤버입니다.
- VM의 경우, 샘플 프로그램은 제품 디스크에 있습니다.
- VSE의 경우, 샘플 프로그램은 QMF 서브라이브러리에 있으며 DSQABFAC.Z로 명명됩니다.

어셈블러 호출 인터페이스의 샘플 프로그램은 다음과 같은 기능을 수행합니다.

- QMF를 시작합니다.
- 세 개의 글로벌 변수를 설정합니다.
- Q1이라는 조회를 실행합니다.
- 서식 F1을 사용하여 결과 보고서를 인쇄합니다.
- QMF 세션을 종료합니다.

QMF는 조회 Q1 또는 서식 F1을 제공하지 않으나, 샘플 프로그램은 이 오브젝트를 사용합니다.

이 절에서는 호출 인터페이스를 사용하여 어셈블러 프로그램을 어셈블, 링크 편집 및 실행하는 방법도 보여줍니다. QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

CICS/MVS 및 CICS/VSE용 샘플 어셈블러 프로그램

DSQABFAC 프로그램은 CICS용 QMF와 함께 제공됩니다.

```

          TITLE 'Sample HLASM Query Callable Interface'
          *****
          *
          * Sample Program: DSQABFAC
          * Assembler Version of the SAA Query Callable Interface
          *
          *****
          DSQABFAC DFHEIENT CODEREG=(12),DATAREG=(13),EIBREG=(11)
          SPACE 1
          *****
          * Start a query interface session
          *****
          LA R4,CICOMM          ESTABLISH ACCESS TO DSQCOMM
          USING DSQCOMM,R4
          SPACE 1
          MVC DSQ_COMM_LEVEL,DSQ_CURRENT_COMM_LEVEL
          ST R4,QMFP1          Address of DSQCOMM
          LA R1,STARTQIL      Address of START command length
          ST R1,QMFP2
          LA R1,STARTQI      Address of START command
          ST R1,QMFP3
          LA R1,1             One Start command parameter
          ST R1,NUMPARMS
          LA R1,NUMPARMS      Address of number of parameters
          ST R1,QMFP4
          LA R1,STARTKYL      Address of keyword lengths
          ST R1,QMFP5
          LA R1,STARTKY      Address of keywords
          ST R1,QMFP6
          LA R1,STARTVL      Address of value lengths
          ST R1,QMFP7
          LA R1,STARTV      Address of values
          ST R1,QMFP8
          LA R1,DSQ_VARIABLE_CHAR Address of value data type
          ST R1,QMFP9
          OI QMFP9,X'80'      Set end of parameter list
          LA R1,QMFP8
          CALL DSQCIA
          SPACE 1

```

그림 26. CICS/MVS 및 CICS/VSE용 DSQABFAC(샘플 HLASM 프로그램) (1/5)

호출 인터페이스 샘플

```

***** 00040000
* Set numeric values into query using SET command * 00041000
***** 00042000
      SPACE 1 00043000
      LA R1,20 Set values for SET GLOBAL command 00044000
      ST R1,VVAL1 00045000
      LA R1,40 00046000
      ST R1,VVAL2 00047000
      LA R1,84 00048000
      ST R1,VVAL3 00049000
      LA R1,SETGL Addr of SET GLOBAL command length 00050000
      ST R1,QMFP2 00051000
      LA R1,SETG Address of SET GLOBAL command 00052000
      ST R1,QMFP3 00053000
      LA R1,3 Three SET GLOBAL variables 00054000
      ST R1,NUMPARMS 00055000
      LA R1,NUMPARMS Address of number of parameters 00056000
      ST R1,QMFP4 00057000
      LA R1,VNAME1L Address of variable name lengths 00058000
      ST R1,QMFP5 00059000
      LA R1,VNAME1 Address of variable names 00060000
      ST R1,QMFP6 00061000
      LA R1,VVAL1L Address of value lengths 00062000
      ST R1,QMFP7 00063000
      LA R1,VVAL1 Address of values 00064000
      ST R1,QMFP8 00065000
      LA R1,DSQ_VARIABLE_FINT Address of value data type 00066000
      ST R1,QMFP9 00067000
      OI QMFP9,X'80' Set end of parameter list 00068000
      LA R1,QMFP9LIST Address of parameter list 00069000
      CALL DSQCIA 00070000
      SPACE 1 00071000
***** 00072000
* Run a query * 00073000
***** 00074000

```

그림 26. CICS/MVS 및 CICS/VSE용 DSQABFAC(샘플 HLASM 프로그램) (2/5)

```

LA R1,QUERYL          Addr of RUN QUERY command length  00075000
ST R1,QMFP2           00076000
LA R1,QUERY           Address of RUN QUERY command  00077000
ST R1,QMFP3           00078000
OI QMFP3,X'80'        Set end of parameter list    00079000
LA R1,QMFPLIST        Address of parameter list    00080000
CALL DSQCIA           00081000
SPACE 1               00082000
***** 00083000
* Print the result of the query * 00084000
***** 00085000
LA R1,REPTL          Addr of PRINT Report command lth  00086000
ST R1,QMFP2           00087000
LA R1,REPT           Address of PRINT Report command  00088000
ST R1,QMFP3           00089000
OI QMFP3,X'80'        Set end of parameter list    00090000
LA R1,QMFPLIST        Address of parameter list    00091000
CALL DSQCIA           00092000
SPACE 1               00093000
***** 00094000
* End the query interface session * 00095000
***** 00096000
LA R1,ENDQIL          Address of EXIT command length  00097000
ST R1,QMFP2           00098000
LA R1,ENDQI           Address of EXIT command  00099000
ST R1,QMFP3           00100000
OI QMFP3,X'80'        Set end of parameter list    00101000
LA R1,QMFPLIST        Address of parameter list    00102000
CALL DSQCIA           00103000
SPACE 1               00104000
***** 00105000
* Return * 00106000
***** 00107000
SPACE 1               00108000

```

그림 26. CICS/MVS 및 CICS/VSE용 DSQABFAC(샘플 HLASM 프로그램) (3/5)

호출 인터페이스 샘플

```

XR      R15,R15          ZERO RETURN CODE          00109000
DFHEIRET RCREG=15      00110000
*****
* Data Areas *
*****
SPACE 1 00112000
* Query Interface commands *
SPACE 1 00113000
STARTQI DC C'START'          START FUNCTION 00114000
SETG DC C'SET GLOBAL'      SET GLOBAL FUNCTION 00115000
QUERY DC C'RUN QUERY Q1'   RUN QUERY 00116000
REPT DC C'PRINT REPORT (FORM=F1)' PRINT REPORT 00117000
ENDQI DC C'EXIT'          END INTERFACE 00118000
SPACE 1 00119000
DS 0F 00120000
STARTQIL DC AL4(L'STARTQI)  LENGTH OF START FUNCTION 00121000
SETGL DC AL4(L'SETG)       LENGTH OF SET GLOBAL FUNCTION 00122000
QUERYL DC AL4(L'QUERY)     LENGTH OF RUN QUERY COMMAND 00123000
REPTL DC AL4(L'REPT)       LENGTH OF PRINT REPORT COMMAND 00124000
ENDQIL DC AL4(L'ENDQI)     LENGTH OF END INTERFACE COMMAND 00125000
SPACE 1 00126000
* START command keyword 00127000
SPACE 1 00128000
STARTKY DC C'DSQSMODE'     00129000
STARTV DC C'INTERACTIVE'   00130000
DS 0F 00131000
STARTKYL DC AL4(L'STARTKY) 00132000
STARTVL DC AL4(L'STARTV)   00133000
SPACE 1 00134000
* SET GLOBAL command variable names 00135000
SPACE 1 00136000
VNAME1 DC C'MYVAR01'       00137000
VNAME2 DC C'SHORT'         00138000
VNAME3 DC C'MYVAR03'       00139000
DS 0F 00140000
VNAME1L DC AL4(L'VNAME1)   00141000
VNAME2L DC AL4(L'VNAME2)   00142000
VNAME3L DC AL4(L'VNAME3)   00143000
SPACE 1 00144000
* SET GLOBAL command values 00145000
SPACE 1 00146000
VVAL1L DC AL4(L'VVAL1)     00147000
VVAL2L DC AL4(L'VVAL2)     00148000
VVAL3L DC AL4(L'VVAL3)     00149000
* Callable interface communications definition 00150000
DSQCOMMA 00151000
00152000
00153000
00154000

```

그림 26. CICS/MVS 및 CICS/VSE용 DSQABFAC(샘플 HLASM 프로그램) (4/5)


```

* Equates for registers 0-15                                00155000
R0      EQU  00                                           00156000
R1      EQU  01                                           00157000
R2      EQU  02                                           00158000
R3      EQU  03                                           00159000
R4      EQU  04                                           00160000
R5      EQU  05                                           00161000
R6      EQU  06                                           00162000
R7      EQU  07                                           00163000
R8      EQU  08                                           00164000
R9      EQU  09                                           00165000
R10     EQU  10                                           00166000
R11     EQU  11                                           00167000
R12     EQU  12                                           00168000
R13     EQU  13                                           00169000
R14     EQU  14                                           00170000
R15     EQU  15                                           00171000
* Local variables located in CICS working storage
DFHEISTG DSECT                                           00172000
          ORG  DFHEIUSR                                   00174000
NUMPARMS DS  F          NUMBER OF KEYWORDS              00175000
* QMF SET GLOBAL command values
VVAL1    DS  F                                           00177000
VVAL2    DS  F                                           00178000
VVAL3    DS  F                                           00179000
* QMF Callable interface parameter list
QMFPLIST DS  0D                                           00181000
QMFP1    DS  F                                           00182000
QMFP2    DS  F                                           00183000
QMFP3    DS  F                                           00184000
QMFP4    DS  F                                           00185000
QMFP5    DS  F                                           00186000
QMFP6    DS  F                                           00187000
QMFP7    DS  F                                           00188000
QMFP8    DS  F                                           00189000
QMFP9    DS  F                                           00190000
* Callable interface communications area
CICOMM   DS  CL(DSQCOMM_LEN)                             00192000
          CSECT
          SPACE 1                                         00193000
          END  DSQABFAC                                   00195000

```

그림 26. CICS/MVS 및 CICS/VSE용 DSQABFAC(샘플 HLASM 프로그램) (5/5)

TSO 및 CMS용 샘플 어셈블리 프로그램

TSO 및 CMS의 경우 QMF는 다음 프로그램과 함께 제공됩니다. 이 프로그램은 DSQABFA로 명명됩니다.

호출 인터페이스 샘플

```

DSQABFA TITLE 'SAMPLE SAA QUERY CALLABLE INTERFACE'
DSQABFA CSECT
*****
*
* Sample Program: DSQABFA
* Assembler Version of the SAA Query Callable Interface
*
*****
        SPACE 1
        STM R14,R12,12(R13)      SAVE ENTRY REGISTERS
        BALR R12,0              INITIALIZE BASE REGISTER
        USING *,R12
        LA R2,SAVEAREA          CHAIN SAVE AREAS
        ST R2,8(R13)
        ST R13,SAVEAREA+4
        LR R13,R2                ESTABLISH SAVE AREA
        SPACE 1
*****
* Start a query interface session
*****
        LA R4,CICOMM            ESTABLISH ACCESS TO DSQCOMM
        USING DSQCOMM,R4
        SPACE 1
        MVC DSQ_COMM_LEVEL,DSQ_CURRENT_COMM_LEVEL
        LA R1,1                  1 PARAMETER
        ST R1,NUMPARMS
        CALL DSQCIA,
            (CICOMM,              QI COMMON AREA
             STARTQIL,            START COMMAND LENGTH
             STARTQI,            START COMMAND
             NUMPARMS,           NUMBER OF KEYWORDS
             STARTKYL,           KEYWORD LENGTHS
             STARTKY,            KEYWORDS
             STARTVL,           VALUE LENGTHS
             STARTV,            VALUES
             DSQ_VARIABLE_CHAR),VL VALUES ARE CHARACTERS
        SPACE 1

```

그림 27. TSO 및 CMS용 DSQABFA(샘플 어셈블러 프로그램) (1/4)

```

*****
* Set numeric values into query using SET command *
*****
SPACE 1
LA R1,20 SET VALUES TO BE MODIFIED
ST R1,VVAL1
LA R1,40
ST R1,VVAL2
LA R1,84
ST R1,VVAL3
LA R1,3 3 PARAMETERS
ST R1,NUMPARMS
SPACE 1
CALL DSQCIA, X
(CICOMM, X
SETGL, SET GLOBAL COMMAND LENGTH X
SETG, SET GLOBAL COMMAND X
NUMPARMS, NUM OF VARIABLES TO BE SET X
VNAME1L, VARIABLE NAME LENGTHS X
VNAME1, VARIABLE NAMES X
VVAL1L, VALUE LENGTHS X
VVAL1, VALUES X
DSQ_VARIABLE_FINT),VL VALUES ARE INTEGERS
SPACE 1
*****
* Run a query *
*****
SPACE 1
CALL DSQCIA, X
(CICOMM, X
QUERYL, QUERY COMMAND LENGTH X
QUERY),VL TEXT OF QUERY COMMAND
SPACE 1
*****
* Print the result of the query *
*****
SPACE 1
CALL DSQCIA,(CICOMM,REPTL,REPT),VL
SPACE 1
*****
* End the query interface session *
*****
SPACE 1
CALL DSQCIA,(CICOMM,ENDQIL,ENDQI),VL
SPACE 1

```

그림 27. TSO 및 CMS용 DSQABFA(샘플 어셈블러 프로그램) (2/4)

호출 인터페이스 샘플

```

*****
* Return
*****
        SPACE 1
        SR    R15,R15                SET RETURN CODE
        L     R13,4(R13)
        L     R14,12(R13)           RESTORE CALLER REGISTERS
        LM    R0,R12,20(R13)
        BR    R14
        EJECT
*****
* Data Areas
*****
        SPACE 1
* Query Interface commands
        SPACE 1
STARTQI DC    C'START'              START FUNCTION
SETG    DC    C'SET GLOBAL'        SET GLOBAL FUNCTION
QUERY   DC    C'RUN QUERY Q1'      RUN QUERY
REPT    DC    C'PRINT REPORT (FORM=F1)' PRINT REPORT
ENDQI   DC    C'EXIT'              END INTERFACE
        SPACE 1
        DS    0F
STARTQIL DC AL4(L'STARTQI)          LENGTH OF START FUNCTION
SETGL    DC    AL4(L'SETG)          LENGTH OF SET GLOBAL FUNCTION
QUERYL   DC    AL4(L'QUERY)        LENGTH OF RUN QUERY COMMAND
REPTL    DC    AL4(L'REPT)          LENGTH OF PRINT REPORT COMMAND
ENDQIL   DC    AL4(L'ENDQI)        LENGTH OF END INTERFACE COMMAND
        SPACE 1
* START command keyword
        SPACE 1
STARTKY  DC    C'DSQSMODE'
STARTV   DC    C'INTERACTIVE'
        DS    0F
STARTKYL DC AL4(L'STARTKY)
STARTVL  DC    AL4(L'STARTV)
        SPACE 1

```

그림 27. TSO 및 CMS용 DSQABFA(샘플 어셈블러 프로그램) (3/4)

```

* SET GLOBAL command variable names
  SPACE 1
VNAME1 DC C'MYVAR01'
VNAME2 DC C'SHORT'
VNAME3 DC C'MYVAR03'
      DS 0F
VNAME1L DC AL4(L'VNAME1)
VNAME2L DC AL4(L'VNAME2)
VNAME3L DC AL4(L'VNAME3)
  SPACE 1
* SET GLOBAL command values
  SPACE 1
VVAL1 DS F
VVAL2 DS F
VVAL3 DS F
VVAL1L DC AL4(L'VVAL1)
VVAL2L DC AL4(L'VVAL2)
VVAL3L DC AL4(L'VVAL3)
  SPACE 1
NUMPARMS DS F NUMBER OF KEYWORDS
  SPACE 1
* callable interface communications area
  SPACE 1
CICOMM DS CL(DSQCOMM_LEN)
  SPACE 1
SAVEAREA DS 18F
      EJECT
      DSQCOMMA
  SPACE 1
R0 EQU 00 EQUATES FOR REGISTERS 0-15
R1 EQU 01
R2 EQU 02
R3 EQU 03
R4 EQU 04
R5 EQU 05
R6 EQU 06
R7 EQU 07
R8 EQU 08
R9 EQU 09
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
  SPACE 1
END DSQABFA

```

그림 27. TSO 및 CMS용 DSQABFA(샘플 어셈블러 프로그램) (4/4)

어셈블리어용 DSQCOMM

이 통신 영역은 버전 2 릴리스 4와 버전 3.2 사이에서 변경되었습니다. QMF는 이 파일을 DSQCOMMMA로서 제공합니다.

```

MACRO                                00001000
DSQCOMMMA                            00002000
*****                              00003000
* Callable Interface - variable constants * 00004000
*****                              00005000
*                                     00006000
* Communications Level ID            00007000
*                                     00008000
DSQ_CURRENT_COMM_LEVEL DC CL12'DSQL>001002<' 00009000
*                                     00010000
* Query Product IDs                  00011000
*                                     00012000
DSQ_QRW DC C'01' 00013000
DSQ_QMF DC C'02' 00014000
DSQ_QM4 DC C'03' 00015000
*                                     00016000
* Query Product Release IDs          00017000
*                                     00018000
DSQ_QRW_V1R2 DC C'01' 00019000
DSQ_QRW_V1R3 DC C'02' 00020000
DSQ_QMF_V2R4 DC C'01' 00021000
DSQ_QMF_V3R1 DC C'02' 00022000
DSQ_QMF_V3R1M1 DC C'03' 00023000
DSQ_QMF_V3R2 DC C'04' 00024000
DSQ_QMF_V3R3 DC C'05' 00025000
DSQ_QMF_V6R1 DC C'06' 00026000
DSQ_QM4_V1R1 DC C'01' 00027000
*                                     00028000
* Extended parameter data types      00029000
*                                     00030000
DSQ_VARIABLE_CHAR DC C'CHAR' 00031000
DSQ_VARIABLE_FINT DC C'FINT' 00032000
*                                     00033000
* Return codes                        00034000
*                                     00035000
DSQ_SUCCESS EQU 0 00036000
DSQ_WARNING EQU 4 00037000
DSQ_FAILURE EQU 8 00038000
DSQ_SEVERE EQU 16 00039000
*                                     00040000
* Instance ID values                  00041000
*                                     00042000
DSQ_CONTINUE EQU 0 00043000
*                                     00044000

```

그림 28. 어셈블러 통신 영역(DSQCOMMMA) (1/2)

```

* Cancel indicator                                00045000
*                                                  00046000
DSQ_CANCEL_YES      EQU  C'1'                    00047000
DSQ_CANCEL_NO       EQU  C'0'                    00048000
*                                                  00049000
*                                                  00050000
DSQ_INTERACTIVE     EQU  C'1'                    00051000
DSQ_BATCH           EQU  C'2'                    00052000
*                                                  00053000
DSQ_YES             EQU  C'1'                    00054000
DSQ_NO              EQU  C'2'                    00055000
*                                                  00056000
*                                                  00057000
*****
* Callable Interface Communications Area          * 00058000
*****                                          00059000
DSQCOMM            DSECT                          00060000
DSQ_RETURN_CODE    DS      F                      FUNCTION RETURN CODE 00061000
DSQ_INSTANCE_ID    DS      F                      ID ESTABLISHED IN START CMD 00062000
DSQ_COMM_LEVEL     DS      CL12                   COMMUNICATIONS LEVEL ID 00063000
DSQ_PRODUCT        DS      CL2                   QUERY PRODUCT ID      00064000
DSQ_PRODUCT_RELEASE DS    CL2                   QUERY PRODUCT RELEASE ID 00065000
DSQ_RESERVE1       DS      CL28                   RESERVED              00066000
DSQ_MESSAGE_ID     DS      CL8                   COMPLETION MESSAGE ID 00067000
DSQ_Q_MESSAGE_ID   DS      CL8                   QUERY MESSAGE ID      00068000
DSQ_START_PARM_ERROR DS    CL8                   START PARAMETER IN ERROR 00069000
DSQ_CANCEL_IND     DS      C                     CMD CANCEL INDICATOR  00070000
DSQ_RESERVE2       DS      CL23                   RESERVED              00071000
DSQ_RESERVE3       DS      CL156                  RESERVED              00072000
DSQ_MESSAGE_TEXT   DS      CL128                  COMPLETION MESSAGE    00073000
DSQ_Q_MESSAGE_TEXT DS    CL128                   QUERY MESSAGE         00074000
      SPACE 1
DSQCOMM_LEN        EQU  *-DSQCOMM                LENGTH OF DSQCOMM AREA 00076000
      MEND                                         00077000

```

그림 28. 어셈블러 통신 영역(DSQCOMMA) (2/2)

CICS에서 어셈블러 프로그램 실행

프로그램을 작성한 후에는 실행하기 전에 프로그램을 변환하고, 어셈블하고, 링크 편집해야 합니다. 이 절에 나열된 예제에서는 반드시 수행해야 할 단계를 보여줍니다. QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

MVS에서 CICS의 변환, 어셈블, 링크 편집

QMF 호출 인터페이스를 사용하는 프로그램을 변환, 어셈블 및 링크 편집할 때, 다음 사항에 유의하십시오.

- 통신 영역 매크로 DSQCOMMA가 어셈블 단계에서 사용가능해야 하며 혹은 프로그램에 DSECT로 복사되어야 합니다.

호출 인터페이스 샘플

- QMF 인터페이스 모듈 DSQCIA가 프로그램의 링크 편집 단계 동안 사용가능해야 합니다.

다음의 JCL은 CICS 제공 프로시저 DFHEBTAL의 예제입니다. 이 프로시저 사용 방법에 대한 지침은 *CICS for VSE/ESA System Definition Guide*를 참조하십시오.

```
//sampasm JOB
//          EXEC PROC=DFHEBTAL
//TRN.SYSIN DD *
*ASM      XOPTS(CICS translator options .....
          .
          Your program or copy of QMF sample DSQABFA
          .
/*
/** Provide Access to QMF Communications Macro DSQCOMM
//ASM.SYSLIB DD DSN=QMF710.SDSQSAPE,DISP=SHR
/** Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF710.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
          INCLUDE CICSLOAD(DFHEAI)
          INCLUDE CICSLOAD(DFHEAI0)
          INCLUDE QMFLOAD(DSQCIA)
          ORDER DFHEAI,DFHEAI0
          ENTRY sampasm
          MODE AMODE(31) RMODE(ANY)
          NAME sampasm(R)
/*
```

그림 29. CICS 변환기, 어셈블러, 연계 편집기를 실행하는 JCL

VSE에서 CICS의 변환, 어셈블, 링크 편집

다음 VSE 작업 제어는 HLASM 프로그램을 VSE/ESA에서 실행되는 CICS로 설치하는 예제입니다. 이 예제는 QMF와 함께 제공되며 DSQ3CIAC.Z라는 이름으로 QMF 서브라이브러리에 들어 있습니다. 자세히 알려면 *CICS for VSE/ESA System Definition Guide* 내용을 참조하십시오.

HLASM이 사용되는 설치 환경이라면 시스템 관리자는 E-Decks의 매크로 처리를 다루는 VSE 라이브러리 종료를 확실히 설정해야 합니다. 이 종료는 DSQCOMMA

를 읽습니다. 이 종료 설정 방법에 대한 설명은 *VSE Guide to System Functions* 및 *IBM High-Level Assembler Programmer's Guide for OS/390, VM and VSE* 내용을 참조하십시오.

프로그램을 어셈블하려면, 다음 HLASM 컴파일러 옵션을 사용하십시오.

```
'LIBMAC,USING(NOLIMIT,NOWARN),EXIT(LIBEXIT(EDECKXIT(ORDER=EA)))'
```

LIBEXIT 매개변수는 CICS 변환 프로세스에 의해 작성된 CICS 매크로 정의를 포함합니다.

```
// JOB DSQ3CIAC
* -----
* Install QMF Callable Interface (HLASM)
* -----
// SETPARM VOLID=valid      *-- update valid for syspch
// SETPARM START=rtrk      *-- update start track/block (syspch)
// SETPARM SIZE=ntrks      *-- update number of tracks/blocks (syspch)
* -----
// DLBL IJSYSPH,'ASM.TRANSLATION',0
// EXTENT SYSPCH,,1,0,&START,&SIZE.
ASSGN SYSPCH,DISK,VOL=&VOLID.,SHR
* Library search chain must contain the QMF, CICS and HLASM sublibrary
// LIBDEF *,SEARCH=(PRD2.PROD,PRD1.BASE,PRD2.CONFIG)
// LIBDEF PHASE,CATALOG=PRD2.PROD
* -----
* Step 1: Translate Callable Interface program
* -----
* You may need to update or remove the SLI statement for your program.
* -----
// EXEC DFHEAP1$
* $$ SLI MEM=DSQABFAC.Z,S=PRD2.PROD
/*
```

그림 30. VSE에서 어셈블러 및 연계 편집기를 실행하기 위한 작업 제어 (1/2)

호출 인터페이스 샘플

```
* -----
* Step 2: Assemble Callable Interface program
* -----
CLOSE SYSPCH,00D
// DLBL IJSYSIN,'ASM.TRANSLATION',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLID.,SHR
// OPTION CATAL,DECK,SYM,ERRS
  PHASE DSQABFAC,*
    INCLUDE DFHEAI
    INCLUDE DFHEAI0
    INCLUDE DSQCIA
    INCLUDE DSQCLOD2
    INCLUDE DSQCMCVP
// EXEC ASMA90,SIZE=(ASMA90,50K), C
      PARM='LIBMAC,USING(NOLIMIT,NOWARN),EXIT(LIBEXIT(EDECKXITC
        (ORDER=EA)))'
CLOSE SYSIPT,SYSRDR
/*
* -----
* Step 3: Link-edit Callable Interface program
* -----
// EXEC LNKEDT,PARM='AMODE=31,RMODE=ANY'
/*
/&
// JOB RESET
ASSGN SYSIPT,SYSRDR IF 1A93D, CLOSE SYSIPT,SYSRDR
ASSGN SYSPCH,00D IF 1A93D, CLOSE SYSPCH,00D
/&
```

그림 30. VSE에서 어셈블러 및 연계 편집기를 실행하기 위한 작업 제어 (2/2)

VM의 CMS에서 프로그램 어셈블 및 실행

다음의 샘플 프로그램은 어셈블러 H 컴파일러를 사용하는 호출 인터페이스 응용 프로그램을 어셈블하고 실행합니다. QMF는 이 예제에서 REXX EXEC를 제공하지 않으나, 여기에 복사하여 설치가 맞게 고칠 수 있습니다.

```

/*****/
/* Assemble your program and execute it. */
/*****/
TRACE off
ADDRESS CMS

/*****/
/* Assemble the program */
/*****/
"ERASE TEMPP MACLIB A"
"MACLIB GEN TEMPP DSQCOMMA"
Maclist = "TEMPP DMSPP CMSLIB OSMACRO"
"GLOBAL MACLIB" Maclist
"HASM yourname"

/*****/
/* Access SQL/DS and initialize database */
/*****/
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"

/*****/
/* Access GDDM product disk */
/*****/
"EXEC PRODUCT GDDM"

/*****/
/* Issue Filedefs for QMF product */
/*****/
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A ( PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 ( PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * ( PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQLDLIB LOADLIB *"

```

그림 31. 프로그램을 어셈블하고 실행하는 REXX 프로그램 (1/2)

```
/* Provide access to QMF and GDDM program libraries */
"GLOBAL LOADLIB DSQDLIB "
"GLOBAL TXTLIB ADMRLIB ADMPLIB ADMGLIB"

Say "Starting to execute 'ASSEMBLER' program"
ADDRESS CMS "RUN yourname"

Exit 0
```

그림 31. 프로그램을 어셈블하고 실행하는 REXX 프로그램 (2/2)

TSO에서 어셈블러 프로그램 실행

프로그램을 어셈블하고 링크 편집한 뒤 TSO에서 프로그램을 실행할 수 있습니다. 다음 절에서는 프로그램을 어셈블하고 링크 편집하는 샘플 작업과 TSO(ISPF 유무에 관계없음)에서 어셈블된 프로그램을 실행하는 샘플 프로그램을 제공합니다.

TSO에서 어셈블 및 링크 편집

다음의 샘플 작업은 어셈블러 H를 사용하여 프로그램을 어셈블하고 링크 편집합니다. 일부 매개변수는 설치에 따라 달라집니다. 자세한 내용은 QMF 관리자에게 문의하십시오.

```

//sampasm JOB
//STEP1 EXEC PROC=ASMHCL
/* Provide Access to QMF Communications Macro DSQCOMM
//C.SYSLIB DD DSN=QMF710.SAMPLIB,DISP=SHR
//C.SYSIN DD *
      .
      Your program or copy of QMF sample DSQABFA
      .
/*
/* Provide Access to QMF Interface Module
//L.QMFLOAD DD DSN=QMF710.SDSQLOAD,DISP=SHR
//L.SYSIN DD *
      INCLUDE QMFLOAD(DSQCIA)
      ENTRY sampasm
      MODE AMODE(31) RMODE(ANY)
      NAME sampasm(R)
/*

```

그림 32. TSO에서 어셈블러 및 연계 편집기를 실행하는 JCL

프로그램을 성공적으로 어셈블한 후에는 이를 실행할 수 있습니다.

ISPF하의 TSO에서 실행

프로그램을 어셈블한 후에는 다음과 유사한 프로그램을 작성하여 ISPF에서 이 프로그램을 실행할 수 있습니다.

```

PROC 0
CONTROL ASIS
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
        DA('QMF710.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD')
ALLOC FI(ISPMLIB) SHR REUSE +
        DA('QMF710.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
        DA('QMF710.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
        DA('QMF710.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
        DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF710.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF710.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(sampasm) NEWAPPL(DSQE)
EXIT CODE(4)

```

그림 33. ISPF하의 TSO에서 프로그램을 실행하는 CLIST

EXIT CODE(4)는 ISPF 특성 패널이 표시되지 않도록 합니다.

ISPF 없이 TSO에서 실행

프로그램을 어셈블한 후에는 다음과 유사한 프로그램을 작성하여 ISPF 없이 TSO에서 이 프로그램을 실행할 수 있습니다.

```

PROC 0
CONTROL ASIS
/*****/
/* Note: QMF, DB2 and GDDM load libraries must be allocated */
/* before executing this CLIST. */
/* Name of QMF load library is "QMF710.SDSQLOAD". */
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF710.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF710.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command */
/*****/
CALL sampasm
EXIT CODE(0)

```

그림 34. ISPF 없이 TSO에서 프로그램을 실행하는 CLIST

C 언어 인터페이스

여기에 제공된 C 언어 호출 인터페이스는 다른 SAA 언어에 대해 제공된 인터페이스와 일치합니다.

C 언어 인터페이스

C 언어의 경우, QMF는 DSQCOMMC 통신 매크로와 DSQCICE 및 DSQCIC의 두 가지 함수 호출을 제공합니다.

주: C++ 응용프로그램으로부터 QMF를 액세스하려면 C 언어로 쓰여진 인터페이스가 필요합니다.

C 언어의 인터페이스 통신 영역 맵핑(DSQCOMMC)

DSQCOMMC는 C 언어에 DSQCOMM 맵핑을 제공하고 해당 제품과 함께 제공됩니다. 표16에서는 DSQCOMMC 값을 보여줍니다.

표 16. DSQCOMMC의 인터페이스 통신 영역

구조 이름	데이터 유형	설명
DSQ_RETURN_CODE	부호 있는 긴 정수	QMF 명령이 실행된 후의 상태를 나타냅니다. 해당 값은 다음과 같습니다. DSQ_SUCCESS 요청의 성공적인 실행 DSQ_WARNING 경고 메시지를 나타내는 정상적인 완료 DSQ_FAILURE 명령이 제대로 실행되지 않음 DSQ_SEVERE 심각한 오류. QMF 세션이 종료됨
DSQ_INSTANCE_ID	부호 있는 긴 정수	START 명령의 실행 동안 QMF에 의해 설정된 ID
DSQ_COMM_LEVEL	문자, 길이 12	DSQCOMM의 레벨을 식별합니다. 이것을 DSQ_CURRENT_COMM_LEVEL 값으로 설정한 다음 QMF START 명령을 발행합니다.
DSQ_PRODUCT	문자, 길이 2	사용중인 IBM 조회 제품을 식별합니다.
DSQ_PRODUCT_RELEASE	문자, 길이 2	사용중인 조회 제품의 릴리스 레벨을 식별합니다.
DSQ_RESERVE1	문자, 길이 28	앞으로 사용되도록 예약됨
DSQ_MESSAGE_ID	문자, 길이 8	완료 메시지 ID
DSQ_Q_MESSAGE_ID	문자, 길이 8	조회 메시지 ID
DSQ_START_PARM_ERROR	문자, 길이 8	START가 매개변수 오류로 인해 실패했을 경우 오류 상태에 있는 매개변수

표 16. DSQCOMM의 인터페이스 통신 영역 (계속)

구조 이름	데이터 유형	설명
DSQ_CANCEL_IND	문자, 길이 1	QMF 명령이 실행되는 동안 사용자가 취소했을 경우 다음 두 값 중 하나를 포함합니다. DSQ_CANCEL_YES DSQ_CANCEL_NO
DSQ_RESERVE2	문자, 길이 23	앞으로 사용되도록 예약됨
DSQ_RESERVE3	문자, 길이 156	앞으로 사용되도록 예약됨
DSQ_MESSAGE_TEXT	문자, 길이 128	완료 메시지 텍스트
DSQ_Q_MESSAGE_TEXT	문자, 길이 128	조회 메시지 텍스트

C 언어의 함수 호출

QMF에서는 C 언어의 두 함수 호출인 DSQCIC 및 DSQCICE를 제공합니다.

DSQCIC

이 호출은 응용프로그램 변수에 액세스를 요구하지 않는 QMF 명령에 대한 것입니다. 대부분의 QMF 명령에서 이 호출을 사용하십시오.

```
DSQCIC (&DSQCOMM,&CMDLTH,&CMDSTR)
```

매개변수 값은 다음과 같습니다.

DSQCOMM

인터페이스 통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이, 긴 유형 매개변수

CMDSTR

실행할 QMF 명령, CMDLTH로 지정된 길이를 가진 부호 없는 문자 유형의 배열로 지정됨. QMF 명령은 대문자여야 합니다.

DSQCICE

이 호출은 응용프로그램 변수에 액세스할 필요없는 세 가지 QMF 명령인 START와 확장 형식 GET GLOBAL 및 SET GLOBAL에 대한 확장 구문으로 되어 있습니다.

```
DSQCICE (&DSQCOMM,&CMDLTH,&CMDSTR,  
&PNUM,&KLTH,&KWORD,  
&VLTH,&VALUE,&VTYPE);
```

매개변수 값은 다음과 같습니다.

DSQCOMM

인터페이스 통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이, 긴 정수 매개변수

CMDSTR

실행할 QMF 명령. 이 명령은 부호 없는 문자 유형의 배열입니다. QMF 명령은 대문자여야 합니다.

PNUM

명령 키워드의 수. 이 값은 긴 정수 매개변수입니다.

KLTH

각 지정된 키워드 &KWORD의 길이. 이 값은 긴 정수 매개변수 또는 긴 정수 매개변수의 배열입니다.

KWORD

QMF 키워드. 각각은 부호 없는 문자 배열입니다.

VLTH

각 키워드와 연관된 값의 길이. 긴 정수 매개변수 또는 긴 정수 매개변수의 배열

VALUE

각 키워드와 연관된 값. 유형은 VTYPE 매개변수에 지정되어 있으며, 유형은 부호 없는 문자 배열이거나, 긴 정수 매개변수이거나, 긴 정수 매개변수의 배열입니다.

VTYPE

값 문자열 VALUE의 데이터 유형. 이 유형은 통신 매크로 DSQCOMMC에서 제공되는 두 값 중 하나입니다.

부호 없는 문자 유형의 경우 DSQ_VARIABLE_CHAR

긴 정수의 경우 DSQ_VARIABLE_FINT

VALUE 필드에서 지정된 모든 값은 VTYPE에서 지정된 데이터 유형을 가집니다.

C 언어 인터페이스는 다른 언어와 유사합니다. 그러나 매개변수에 대해 다음 사항을 고려해야 합니다.

- 명령 문자열, START, GET 및 SET 명령 매개변수는 모두 입력 문자열입니다. 따라서 C를 사용하기 위해서는 널 값으로 종료되는 기억영역을 제공해야 하며, 이때 널 값은 매개변수 길이에 포함됩니다. 컴파일 시간 길이 함수를 사용하여 QMF 인터페이스에 전달된 매개변수 길이를 구해야 합니다.
- 문자열이 문자열의 끝에 도달하기 전에 널에 의해 종료되지 않으면 QMF는 오류를 발생시킵니다. 널 값(X'00')은 문자열의 종료를 나타냅니다.
- GET 명령에 의해 획득된 값을 포함하여 출력 매개변수 문자열에 해당하는 C 매개변수의 경우 QMF는 QMF 기억영역에서 응용프로그램 기억영역까지 데이터를 이동시키며 문자열의 끝에 널 표시기를 설정합니다. 문자열이 사용자의 기억영역에서 맞지 않으면 경고 메시지가 발행하고 데이터는 오른쪽에서 잘립니다. 널 표시기는 항상 데이터 문자열의 끝부분에 위치합니다.

마이그레이션 정보

DSQCOMM은 버전 2 릴리스 4에서 버전 3.2로 변경되었습니다.

- 이전 DSQCOMM을 계속해서 사용하려면 프로그램을 다시 컴파일할 필요가 없습니다.
- DSQCICX의 버전 3.2를 사용하려면 버전 2 릴리스 4 프로그램을 다시 링크 편집해야 합니다.

그러나 버전 3.2 DSQCOMM은 START 명령에 오류가 있는 경우 특히 유용한 메시지 텍스트를 제공합니다. 새 DSQCOMM을 사용하려면, 프로그램을 다시 어셈블한 후 DSQ_COMM_LEVEL(DSQCOMM에서)을 DSQ_CURRENT_COMM_LEVEL로 초기화해야 합니다. 이 값을 설정하지 않으면 QMF는 DSQCOMM을 버전 2 릴리스 4로 처리합니다.

MVS에서 CICS 사용자를 위한 주의사항

DSQCICX는 버전 3 릴리스 1 수정판 1에서 버전 3 릴리스 2로 변경되었습니다. QMF 제공 함수 호출과 주 QMF 프로그램간의 인터페이스는 CALL 인터페이스에서 EXEC CICS LINK 인터페이스로 변경되었습니다. 새 인터페이스는 사용자 프로그램과 QMF 제품을 확실하게 구별해 줍니다. 인터페이스가 변경되었기 때문에 호출 인터페이스를 사용한 프로그램을 다시 링크 편집해야 합니다.

C 언어 프로그래밍 예제

이 예제에서는 IBM C 언어의 SAA 호출 인터페이스를 보여줍니다.

다음 프로그램 DSQABFC는 QMF 제품과 함께 제공됩니다. 여기에서 샘플 소스 코드 목록을 볼 수 있고 온라인으로 액세스할 수도 있습니다.

- OS/390의 경우, 샘플 프로그램은 라이브러리 QMF710.SDSQSAPE의 멤버입니다.
- VM의 경우, 샘플 프로그램은 제품 디스크에 있습니다.
- VSE의 경우, 샘플 프로그램은 QMF 서브라이브러리에 있으며 DSQABFAC.Z로 명명됩니다.

C 언어 호출 인터페이스의 샘플 프로그램은 다음 기능을 수행합니다.

- QMF를 시작합니다.
- 세 개의 글로벌 변수를 설정합니다.
- Q1이라는 조회를 실행합니다.
- 서식 F1을 사용하여 결과 보고서를 인쇄합니다.
- QMF 세션을 종료합니다.

QMF는 조회 Q1 또는 서식 F1을 제공하지 않으나, 샘플 프로그램은 이 오브젝트를 사용합니다.

이 절에서는 호출 인터페이스를 사용하는 C 언어 프로그램을 컴파일하고, 링크 편집하고, 실행하는 방법을 보여줍니다. QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

```

/*****
/* Sample Program: DSQABFC                               */
/* C Version of the SAA Query Callable Interface          */
/*****

/*****
/* Include standard and string "C" functions            */
/*****
#include <string.h>
#include <stdlib.h>

/*****
/* Include and declare query interface communications area */
/*****
#include <DSQCOMM.C.H>

int main()
{

    struct dsqcomm communication_area;    /* DSQCOMM from include */

/*****
/* Query interface command length and commands          */
/*****
signed long command_length;
static char start_query_interface[] = "START";
static char set_global_variables[] = "SET GLOBAL";
static char run_query[] = "RUN QUERY Q1";
static char print_report[] = "PRINT REPORT (FORM=F1)";
static char end_query_interface[] = "EXIT";

/*****
/* Query command extension, number of parameters and lengths */
/*****
signed long number_of_parameters;    /* number of variables */
signed long keyword_lengths[10];    /* lengths of keyword names */
signed long data_lengths[10];    /* lengths of variable data */

```

그림 35. 샘플 C 프로그램(DSQABFC) (1/3)

C 언어 인터페이스

```
/* *****  
/* Variable data type constants */  
/* *****  
static char char_data_type[] = DSQ_VARIABLE_CHAR;  
static char int_data_type[] = DSQ_VARIABLE_FINT;  
  
/* *****  
/* Keyword parameter and value for START command */  
/* *****  
static char start_keywords[] = "DSQSMODE";  
static char start_keyword_values[] = "INTERACTIVE";  
/* *****  
/* Keyword parameter and values for SET command */  
/* *****  
#define SIZE_VAL 8  
char set_keywords [3][SIZE_VAL]; /* Parameter name array */  
signed long set_values[3]; /* Parameter value array */  
  
/* *****  
/* MAIN PROGRAM */  
/* *****  
  
/* *****  
/* Start a Query Interface Session */  
/* *****  
strncpy (communication_area.dsq_comm_level,  
         DSQ_CURRENT_COMM_LEVEL,  
         sizeof(communication_area.dsq_comm_level));  
number_of_parameters = 1;  
command_length = sizeof(start_query_interface);  
keyword_lengths[0] = sizeof(start_keywords);  
data_lengths[0] = sizeof(start_keyword_values);  
dsqciice(&communication_area,  
        &command_length,  
        &start_query_interface[0],  
        &number_of_parameters,  
        &keyword_lengths[0],  
        &start_keywords[0],  
        &data_lengths[0],  
        &start_keyword_values[0],  
        &char_data_type[0]);
```

그림 35. 샘플 C 프로그램(DSQABFC) (2/3)

```

/*****
/* Set numeric values into query using SET command */
/*****
    number_of_parameters = 3;
    command_length = sizeof(set_global_variables);
    strcpy(set_keywords[0], "MYVAR01");
    strcpy(set_keywords[1], "SHORT");
    strcpy(set_keywords[2], "MYVAR03");
    keyword_lengths[0] = SIZE_VAL;
    keyword_lengths[1] = SIZE_VAL;
    keyword_lengths[2] = SIZE_VAL;
    data_lengths[0] = sizeof(long);
    data_lengths[1] = sizeof(long);
    data_lengths[2] = sizeof(long);
    set_values[0] = 20;
    set_values[1] = 40;
    set_values[2] = 84;
    dsqcice(&communication_area,;
            &command_length,;
            &set_global_variables[0],
            &number_of_parameters,;
            &keyword_lengths[0],
            &set_keywords[0][0],
            &data_lengths[0],
            &set_values[0],
            &int_data_type[0]);

/*****
/* Run a Query */
/*****
    command_length = sizeof(run_query);
    dsqic(&communication_area, &command_length,;
          &run_query[0]);

/*****
/* Print the results of the query */
/*****
    command_length = sizeof(print_report);
    dsqic(&communication_area, &command_length,;
          &print_report[0]);

/*****
/* End the query interface session */
/*****
    command_length = sizeof(end_query_interface);
    dsqic(&communication_area, &command_length,;
          &end_query_interface[0]);
    exit(0);
}

```

그림 35. 샘플 C 프로그램(DSQABFC) (3/3)

C용 DSQCOMM

이 포함 파일 DSQCOMM.C는 QMF 제품과 함께 제공됩니다.

```

/*****
/* C Include for Query Callable Interface (MVS/VM) */
/*****

/* Structure declare for Communications Area */

struct dsqcomm {
    long int dsq_return_code; /* Function return code */
    long int dsq_instance id; /* id established in start cmd*/
    char dsq_comm_level[12]; /* communications level id */
    char dsq_product[2]; /* query product id */
    char dsq_product_release[2]; /* query product release */
    char dsq_reserve1[28]; /* reserved */
    char dsq_message_id[8]; /* completion message ID */
    char dsq_q_message_id[8]; /* query message ID */
    char dsq_start_parm_error[8]; /* start parameter in error */
    char dsq_cancel_ind[1]; /* cmd cancelled indicator */
    /* 1 = cancelled, 0 = not cancelled*/
    char dsq_reserve2[23]; /* RESERVED AREAS */
    char dsq_reserve3[156];
    char dsq_message_text[128]; /* Message text */
    char dsq_q_message_text[128]; /* Query message text */
};

/* RETURN CODES */

#define DSQ_SUCCESS 0
#define DSQ_WARNING 4
#define DSQ_FAILURE 8
#define DSQ_SEVERE 16

/* Communications Level */

#define DSQ_CURRENT_COMM_LEVEL "DSQL>001002<"

/* Query Product Codes */

#define DSQ_QRW "01"
#define DSQ_QMF "02"
#define DSQ_QM3 "03"

/* Query Product Release Levels */

#define DSQ_QRW_V1R2 "01"
#define DSQ_QRW_V1R3 "02"
#define DSQ_QMF_V2R4 "01"
#define DSQ_QMF_V3R1 "02"
#define DSQ_QMF_V3R1M1 "03"
#define DSQ_QMF_V3R2 "04"
#define DSQ_QMF_V3R3 "05"
#define DSQ_QMF_V6R1 "06"
#define DSQ_QM4_V1R1 "01"

```

그림 36. C 통신 영역(DSQCOMM.C) (1/2)


```

/* INSTANCE CODES */
#define DSQ_CONTINUE          0
/* CANCELLED INDICATOR */
#define DSQ_CANCEL_YES      "1"
#define DSQ_CANCEL_NO      "0"
/* VARIABLE TYPES */
#define DSQ_VARIABLE_CHAR   "CHAR"
#define DSQ_VARIABLE_FINT   "FINT"
#define DSQ_INTERACTIVE     "1"
#define DSQ_BATCH           "2"
#define DSQ_YES             "1"
#define DSQ_NO              "2"
/* Call Interface structure */
/* Calling format for normal call with 3 parameters */
#define dsqcic(parm1, parm2, parm3 )\
        dsqcicx( parm1, parm2, parm3)
/* Calling format for call with CMD_EXT area 9 parameters */
#define dsqcice(parm1, parm2, parm3,\
        parm4, parm5, parm6, parm7, parm8, parm9 )\
        dsqcicx( parm1, parm2, parm3, \
        parm4, parm5, parm6, \
        parm7, parm8, parm9 )
/* DECLARE OS LINKAGE FORMAT */
#pragma linkage(dsqcicx, OS)

```

그림 36. C 통신 영역(DSQCOMMC) (2/2)

CICS에서 프로그램 실행

프로그램을 작성한 후에는 실행하기 전에 프로그램을 변환, 컴파일 및 링크 편집해야 합니다. 이 절에 나열된 예제에서는 반드시 수행해야 할 단계를 보여줍니다. QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

CICS에서 QMF 호출 인터페이스를 사용하는 프로그램을 변환, 컴파일 및 링크 편집할 경우, 다음 사항을 고려하십시오.

- 통신 영역 매크로 DSQCOMMC를 컴파일 단계에서 사용가능해야 하며 혹은 프로그램에 복사해야 합니다.

C 언어 인터페이스

- QMF 인터페이스 모듈 DSQCICX를 프로그램의 링크 편집 단계 동안 사용 가능해야 합니다.

MVS에서 CICS의 변환, 컴파일, 링크 편집

다음 예제에서는 CICS 제공 프로시저 DFHEBTDL를 사용합니다. 이 프로시저 사용 방법에 대한 지침은 *CICS for VSE/ESA System Definition Guide*를 참조하십시오.

```
//sampleC JOB
// EXEC PROC=DFHEBTDL
//TRN.SYSIN DD *
#pragma XOPTS(CICS translator options .....)
.
Your program or copy of QMF sample DSQABFC
.
/*
/** Provide Access to QMF Communications Macro DSQCOMM
//ASM.SYSLIB DD DSN=QMF710.SDSQSAPE,DISP=SHR
/** Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF710.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
INCLUDE CICSLOAD(DFHELII)
INCLUDE QMFLOAD(DSQCICX)
ORDER DFHELII
ENTRY sampleC
MODE AMODE(31) RMODE(ANY)
NAME sampleC(R)
/*
```

그림 37. CICS 변환기, C 컴파일러 및 연계 편집기를 실행하는 JCL

C/370 언어 프로그램은 AMODE=31로 링크 편집되어야 합니다.

VSE에서 CICS의 변환, 컴파일 및 링크 편집

C/370 사전 링크 단계 동안 PRD2.PROD 서브라이브러리(QMF 기본 설치 서브라이브러리)에 위치한 IBM 제공 인터페이스 오브젝트(DSQCICX.OBJ, DSQCLOD2.OBJ 및 DSQCMCVP.OBJ)를 LIBDEF * 검색 체인에서 사용할 수 있어야 합니다.

183 페이지의 그림 38과 같이 링크 편집 단계 동안 CICS 어셈블러 인터페이스 DFHEAI0는 LIBDEF * 검색 체인에 있어야 합니다.

이 샘플 작업 제어는 PRD2.PROD에 DSQ3CIC.Z로 저장됩니다.

```
// JOB DSQ3CIC      Sample job to Install QMF Callable Interface (C/370)
* -----
* Install QMF Callable Interface Example (C/370)
* -----
// SETPARAM VOLID=volid      *-- update volid for sypch
// SETPARAM START=rtrk      *-- update start track/block
// SETPARAM SIZE=ntrks      *-- update number of tracks/blocks
// SETPARAM VOLID2=volid2    *-- update volid for work area
// SETPARAM START2=rtrk     *-- update start track/block
// SETPARAM SIZE2=ntrks     *-- update number of tracks/blocks
* -----
// DLBL  IJSYSPH,'c.translation',0
// EXTENT SYSPCH,,1,0,&START,&SIZE
ASSGN SYSPCH,DISK,VOL=&VOLID,SHR
* Library search chain must contain the QMF, CICS and C/370 sublibrary
// LIBDEF *,SEARCH=(PRD2.PROD,PRD1.BASE,PRD2.CONFIG)
// LIBDEF PHASE,CATALOG=PRD2.PROD
* -----
* Step 1: Translate callable interface program (C/370)
* -----
* You may need to update or remove the SLI statement for your program.
* -----
// EXEC  DFHEDP1$,SIZE=256K
..* $$ SLI MEM=DSQABFC.Z,S=PRD2.QMFD
/*
CLOSE SYSPCH,00D
* -----
* Step 2: Compile callable interface program (C/370)
* -----
// DLBL  IJSYSIN,'c.translation',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLID,SHR
// DLBL  IJSYSPH,'compiler.output',0
// EXTENT SYSPCH,,1,0,&START2,&SIZE2
ASSGN SYSPCH,DISK,VOL=&VOLID2,SHR
// OPTION DECK
// EXEC  EDCCOMP,SIZE=EDCCOMP,PARM='RENT'
CLOSE SYSIPT,SYSRDR
CLOSE SYSPCH,00D
```

그림 38. CICS/VSE 변환기, C 컴파일러 및 연계 편집기를 실행하는 작업 제어 (1/2)

C 언어 인터페이스

```
* -----  
* Step 3: Pre-link callable interface program (C/370)  
* -----  
// DLBL IJSYSIN,'compiler.output',0  
// EXTENT SYSIPT  
ASSGN SYSIPT,DISK,VOL=&VOLID2,SHR  
// OPTION CATAL,NODECK  
    PHASE DSQABFC,*  
        INCLUDE DFHELII  
        INCLUDE DFHEAI0  
// EXEC EDCPRLK,SIZE=EDCPRLK  
CLOSE SYSIPT,SYSRDR  
/*  
* -----  
* Step 4: Link-edit callable interface program (C/370)  
* -----  
// EXEC LNKEDT,PARM='AMODE=24,RMODE=24'  
/*  
/&  
// JOB RESET  
ASSGN SYSIPT,SYSRDR    IF 1A93D, CLOSE SYSIPT,SYSRDR  
ASSGN SYSPCH,00D      IF 1A93D, CLOSE SYSPCH,00D  
/&
```

그림 38. CICS/VSE 변환기, C 컴파일러 및 연계 편집기를 실행하는 작업 제어 (2/2)

VM의 CMS에서 프로그램 컴파일 및 실행

다음 프로그램은 IBM C 컴파일러를 사용하여 호출 인터페이스를 컴파일하고 실행합니다.

QMF는 이 예제에서 REXX EXEC를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

```

/*****/
/* Compile your program and run it. */
/*****/
TRACE off
ADDRESS CMS
/*****/
/* Access C product disk using an exec, PRODUCT, that you write. */
/*****/
EXEC PRODUCT ADC370
/*****/
/* Compile the program */
/*****/
"GLOBAL TXTLIB IMBLIB SCEELKED"
"GLOBAL LOADLIB EDCLINK SCREERUN"
"CC" PNAME "(SOURCE SHOWINC"
/*****/
/* Create an executable "C" module file */
/*****/
"GLOBAL LOADLIB DSQDLIB SCREERUN"
"GLOBAL TXTLIB EDCBASE ADMRLIB ADMPLIB ADMGLIB"
"CMOD yourname DSQCICX DSQCLOD2 DSQCMCVP"
/*****/
/* Access SQL/DS and initialize database */
/*****/
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"
/*****/
/* Access GDDM product disk */
/*****/
"EXEC PRODUCT GDDM"
/*****/
/* Issue Filedefs for QMF product */
/*****/
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A ( PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 ( PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * ( PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQDLIB LOADLIB *"
/*****/
/* Provide access to QMF and C program libraries */
/*****/
"GLOBAL LOADLIB DSQDLIB SCREERUN"
"GLOBAL TXTLIB EDCBASE ADMRLIB ADMPLIB ADMGLIB"
Say "Starting to run 'C' program"
"yourname"

Exit 0

```

그림 39. 프로그램을 컴파일하고 실행하는 REXX 프로그램

설치를 알맞게 이 프로그램을 변경해야 합니다.

TSO에서 C 프로그램 실행

다음 절은 호출 인터페이스를 컴파일하고, 링크 편집하기 위한 샘플 작업과 컴파일된 프로그램(ISPF 유무에 관계없음)을 실행하기 위한 샘플 프로그램을 제공합니다.

TSO에서 컴파일 및 링크 편집

다음 작업은 MVS용 IBM C 컴파일러를 사용하여 호출 인터페이스 응용프로그램을 컴파일하고 링크 편집합니다. 일부 매개변수의 경우 설치에 따라 달라질 수 있습니다. 자세한 내용은 QMF 관리자에게 문의하십시오.

```
//sampleC    JOB
//STEP1      EXEC PROC=EDCCL,LPARM='MAP'
/** Provide Access to QMF Communications Macro DSQCOMM
//COMPILE.SYSLIB DD DSN=QMF710.SAMPLIB,DISP=SHR
//COMPILE.SYSIN DD DATA,DLM='<>'

      .
      Your program or copy of QMF sample DSQABFC
      .

<>
/** Provide Access to QMF Interface Module DSQCICX
//LKED.SYSLIB DD DSN=QMF710.SDSQLOAD,DISP=SHR
/*
```

그림 40. TSO에서 C 컴파일러 및 연계 편집기를 실행하는 JCL

ISPF 없이 TSO에서 프로그램 실행

프로그램을 컴파일한 후에는 다음과 유사한 프로그램을 작성하여 이를 실행할 수 있습니다.

```

PROC 0
CONTROL ASIS
/*****
/* Note: QMF, DB2, GDDM and C load libraries must be      */
/*      allocated before running this CLIST.              */
/*      Name of QMF load library is "QMF710.SDSQLOAD".    */
/*****
/* Specify attribute list for dataset allocations          */
/*****
ATTR PRINTDCB  LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB  LRECL(80)  RECFM(F B)   BLKSIZE(3120)
ATTR UDUMPDCB  LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB   LRECL(79)  RECFM(F B A) BLKSIZE(4029)
/*****
/* Datasets used by TSO                                  */
/*****
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
/*****
/* QMF/GDDM Datasets                                    */
/*****
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQUCFRM) DA('QMF710.DSQUCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM')   SHR REUSE
ALLOC FI(ADMGDF)   DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS)  DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****
/* Datasets used by QMF                                  */
/*****
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP)  SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQPILL)  NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT)  NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE)  DA('QMF710.DSQPNLE') SHR REUSE
/*****
/* Start your program using TSO CALL command            */
/*****
CALL sampleC
EXIT CODE(0)

```

그림 41. ISPF 없이 TSO에서 프로그램을 실행하는 CLIST

ISPF하의 TSO에서 프로그램 실행

프로그램을 컴파일한 후에는 다음과 유사한 프로그램을 작성하여 이를 실행할 수 있습니다.

```
PROC 0
CONTROL ASIS
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
```

그림 42. ISPF하의 TSO에서 DSQABFC를 실행하는 CLIST (1/2)


```

/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +

      DA('QMF710.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD', +
      'EDC.SEDCLINK','PLI.SIBMLINK')
ALLOC FI(ISPMLIB) SHR REUSE +
      DA('QMF710.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
      DA('QMF710.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
      DA('QMF710.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
      DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF710.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF710.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(sampleC) NEWAPPL(DSQE)
EXIT CODE(4)

```

그림 42. ISPF하의 TSO에서 DSQABFC를 실행하는 CLIST (2/2)

EXIT CODE(4)는 ISPF 특성 패널이 표시되지 않도록 합니다.

s

COBOL 언어 인터페이스

여기에 제공된 COBOL 호출 인터페이스는 다른 SAA 언어에 대해 제공된 것과 일치합니다.

COBOL로 호출 인터페이스 프로그램을 작성하려면 VS COBOL II, COBOL/370, MVS 및 VM용 IBM COBOL, VSE용 IBM COBOL을 사용해야 합니다. ³

COBOL의 인터페이스 통신 영역 맵핑(DSQCOMMB)

DSQCOMMB는 COBOL에 대해 DSQCOMM 맵핑을 제공하고 해당 제품과 함께 제공됩니다. 표17은 DSQCOMMB의 값입니다.

표 17. DSQCOMMB의 인터페이스 통신 영역

구조 이름	데이터 유형	설명
DSQ-RETURN-CODE	PIC 9(8)	QMF 명령이 실행된 후의 상태를 나타냅니다. 해당 값은 다음과 같습니다. DSQ-SUCCESS 요청의 성공적인 실행 DSQ-WARNING 경고 메시지를 나타내는 정상적인 완료 DSQ-FAILURE 명령이 제대로 실행되지 않음 DSQ-SEVERE 심각한 오류. QMF 세션이 종료됨
DSQ-INSTANCE-ID	PIC 9(8)	START 명령의 실행 동안 QMF에 의해 설정된 ID
DSQ-COMM-LEVEL	PIC X(12)	DSQCOMM의 레벨을 식별합니다. 이것을 DSQ_CURRENT_COMM_LEVEL 값으로 설정한 다음 QMF START 명령을 발행합니다.
DSQ-PRODUCT	PIC X(2)	사용중인 IBM 조희 제품을 식별합니다.
DSQ-PRODUCT-RELEASE	PIC X(2)	사용중인 조희 제품의 릴리스 레벨을 식별합니다.
DSQ-RESERVE1	PIC X(28)	앞으로 사용되도록 예약됨
DSQ-MESSAGE-ID	PIC X(8)	완료 메시지 ID
DSQ-Q-MESSAGE-ID	PIC X(8)	조희 메시지 ID
DSQ-START-PARM-ERROR	PIC X(8)	START가 매개변수 오류로 인해 실패했을 경우 오류 상태에 있는 매개변수

3. COBOL/370은 CICS/VSE에서 지원되지 않습니다.

표 17. DSQCOMM의 인터페이스 통신 영역 (계속)

구조 이름	데이터 유형	설명
DSQ-CANCEL-IND	PIC X(1)	QMF 명령이 실행되는 동안 사용자가 취소했을 경우 다음 두 값 중 하나를 포함합니다. DSQ-CANCEL-YES DSQ-CANCEL-NO
DSQ-RESERVE2	PIC X(23)	앞으로 사용되도록 예약됨
DSQ-RESERVE3	PIC X(156)	앞으로 사용되도록 예약됨
DSQ-MESSAGE-TEXT	PIC X(128)	완료 메시지 텍스트
DSQ-Q-MESSAGE-TEXT	PIC X(128)	조회 메시지 텍스트

COBOL의 함수 호출

QMF는 COBOL 언어에 대해 하나의 함수 호출인 DSQCIB를 제공합니다. 이것은 통신 매크로 DSQCOMM에 설명되어 있습니다. 이 함수 호출에는 DSQCIB 및 DSQCIB 확장 형식의 두 가지 형식이 있습니다.

DSQCIB

이 호출은 응용프로그램 변수에 액세스를 요구하지 않는 QMF 명령에 대한 것입니다. 대부분의 QMF 명령에서 이 호출을 사용하십시오.

```
CALL DSQCIB USING DSQCOMM CMDLTH CMDSTR
```

매개변수 값은 다음과 같습니다.

DSQCOMM

인터페이스 통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이. 이 값은 정수 매개변수입니다.

CMDSTR

실행할 QMF 명령. 이 값은 CMDLTH에 의해 지정된 길이의 대문자 문자열입니다.

DSQCIB, 확장 형식

이 호출은 응용프로그램 변수에 액세스할 필요없는 세 가지 QMF 명령인 START와 확장 형식 GET GLOBAL 및 SET GLOBAL에 대한 확장 구문으로 되어 있습니다.

```
DSQCIB USING  
DSQCOMM CMDLTH CMDSTR  
PNUM KLTH KWORD VLTH VALUE VTYPE
```

매개변수 값은 다음과 같습니다.

DSQCOMM

인터페이스 통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이. 이 값은 정수 매개변수입니다.

CMDSTR

실행할 QMF 명령. 이 값은 CMDLTH에 의해 지정된 길이의 대문자 문자열입니다.

PNUM

명령 키워드의 수. 이 값은 정수 매개변수입니다.

KLTH

각 지정된 키워드의 길이. 이 값은 정수 매개변수 또는 정수 매개변수의 배열입니다.

KWORD

QMF 키워드. 각각은 문자 또는 KLTH에 의해 지정된 것과 같은 길이를 가지는 문자 또는 문자의 구조입니다. 모든 키워드가 같은 길이이면 문자의 배열을 사용할 수 있습니다.

VLTH

키워드와 연관된 각 값의 길이. 이 값은 정수 매개변수 또는 정수 매개변수의 배열입니다.

VALUE

각 키워드와 연관된 값. 유형은 VTYPE 매개변수에서 지정되어 있으며 문자, 문자의 구조, 정수 매개변수 또는 정수 매개변수의 배열이 될 수 있습니다.

VTTYPE

값 문자열 VALUE의 QMF 데이터 유형. 이 유형은 통신 매크로 DSQCOMM에서 제공되는 두 값 중 하나입니다.

문자 값의 경우 DSQ-VARIABLE-CHAR

정수 값의 경우 DSQ-VARIABLE-FINT

VALUE 필드에서 지정된 모든 값은 VTYPE에서 지정된 데이터 유형을 가집니다.

COBOL에서 ISPF LIBDEF 서비스 사용

QMF 인터페이스 DSQCIB에 대한 동적 호출을 사용중이고 QMF 응용프로그램에서 LIBDEF 함수를 사용하려면 동적 호출을 정적 호출로 변경하십시오. 예를 들면 다음 호출 ID 명령문을

```
CALL DSQCIB USING ...
```

다음과 같은 해당 호출 리터럴 서식으로 변경하십시오.

```
CALL "DSQCIB" USING ...
```

마이그레이션 정보

DSQCOMM은 버전 2 릴리스 4에서 버전 3.2로 변경되었습니다.

- 이전 DSQCOMM을 계속해서 사용하려면 프로그램을 다시 컴파일할 필요가 없습니다.
- DSQCIB의 버전 3.2를 사용하려면 버전 2 릴리스 4 프로그램을 다시 링크 편집해야 합니다.

새 DSQCOMM는 START 명령에 오류가 있는 경우 특히 유용한 메시지 텍스트를 제공합니다. 새 DSQCOMM을 사용하려면, 프로그램을 다시 어셈블한 후 DSQ_COMM_LEVEL(DSQCOMM에서)을 DSQ_CURRENT_COMM_LEVEL로 초기화해야 합니다. 이 값을 설정하지 않으면 QMF는 DSQCOMM을 버전 2 릴리스 4로 처리합니다.

MVS에서 CICS의 사용자를 위한 주의사항

DSQCIB는 버전 3 릴리스 1 수정판 1에서 버전 3 릴리스 2로 변경되었습니다. QMF 제품 함수 호출과 주 QMF 프로그램간의 인터페이스는 CALL 인터페이스에서 EXEC CICS LINK 인터페이스로 변경되었습니다. 새 인터페이스는 사용자 프로그램과 QMF 제품을 확실하게 구별해 줍니다. 인터페이스가 변경되었기 때문에 호출 인터페이스를 사용하는 프로그램을 다시 링크 편집해야 합니다.

COBOL 프로그래밍 예제

다음 프로그램인 DSQABFCO는 QMF 제품과 함께 제공됩니다. 이 예제는 VS COBOL II를 사용합니다.

여기에서 샘플 소스 코드 목록을 볼 수 있고 온라인으로 액세스할 수도 있습니다.

- OS/390의 경우, 샘플 프로그램은 라이브러리 QMF710.SDSQSAPE의 멤버입니다.
- VM의 경우, 샘플 프로그램은 제품 디스크에 있습니다.
- VSE의 경우, 샘플 프로그램은 QMF 서브라이브러리에 위치하며 DSQABFCO.Z로 명명됩니다.

COBOL 호출 인터페이스의 샘플 프로그램은 다음 기능을 수행합니다.

- QMF를 시작합니다.
- 세 개의 글로벌 변수를 설정합니다.
- Q1이라는 조회를 실행합니다.
- 서식 F1을 사용하여 결과 보고서를 인쇄합니다.
- QMF 세션을 종료합니다.

QMF는 조회 Q1 또는 서식 F1을 제공하지 않으나, 샘플 프로그램은 이 오브젝트를 사용합니다.

이 절에서는 호출 인터페이스를 사용하여 COBOL 프로그램을 컴파일하고, 링크 편집하고, 실행하는 방법을 보여줍니다. QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

```

*****
* The following is a VS COBOL II version of the query
* callable interface *** DSQABFCO **.
*****
IDENTIFICATION DIVISION.
PROGRAM-ID. DSQABFCO.
DATE-COMPILED.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*****
* Copy DSQCOMMB definition - contains query interface variables
*****
COPY DSQCOMMB.

* Query interface commands
01 STARTQI PIC X(5) VALUE "START".
01 SETG PIC X(10) VALUE "SET GLOBAL".
01 QUERY PIC X(12) VALUE "RUN QUERY Q1".
01 REPT PIC X(22) VALUE "PRINT REPORT (FORM=F1 ".
01 ENDQI PIC X(4) VALUE "EXIT".

* Query command length
01 QICLTH PIC 9(8) USAGE IS COMP-4.
* Number of variables
01 QIPNUM PIC 9(8) USAGE IS COMP-4.
* Keyword variable lengths
01 QIKLTHS.
03 KLTHS PIC 9(8) OCCURS 10 USAGE IS COMP-4.
* Value Lengths
01 QIVLTHS.
03 VLTHS PIC 9(8) OCCURS 10 USAGE IS COMP-4.
* Start Command Keyword
01 SNAMEs.
03 SNAME1 PIC X(8) VALUE "DSQSMODE".
* Start Command Keyword Value
01 SVALUES.
03 SVALUE1 PIC X(11) VALUE "INTERACTIVE".
* Set GLOBAL Command Variable Names to set
01 VNAMEs.
03 VNAME1 PIC X(7) VALUE "MYVAR01".
03 VNAME2 PIC X(5) VALUE "SHORT".
03 VNAME3 PIC X(7) VALUE "MYVAR03".
* Variable value parameters
01 VVALUES.
03 VVALS PIC 9(8) OCCURS 10 USAGE IS COMP-4.

01 TEMP PIC 9(8) USAGE IS COMP-4.

```

그림 43. 샘플 COBOL 프로그램(DSQABFCO) (1/2)

COBOL 언어 인터페이스

```
PROCEDURE DIVISION.  
*  
* Start a query interface session  
  MOVE DSQ-CURRENT-COMM-LEVEL TO DSQ-COMM-LEVEL.  
  MOVE 5 TO QICLTH.  
  MOVE 8 TO KLTHS(1).  
  MOVE 11 TO VLTHS(1).  
  MOVE 1 TO QIPNUM.  
  CALL DSQCIB USING DSQCOMM, QICLTH, STARTQI,  
                  QIPNUM, QIKLTHS, SNAMES,  
                  QIVLTHS, SVALUES, DSQ-VARIABLE-CHAR.  
*  
* Set numeric values into query variables using SET GLOBAL command  
  MOVE 10 TO QICLTH.  
  MOVE 7 TO KLTHS(1).  
  MOVE 5 TO KLTHS(2).  
  MOVE 7 TO KLTHS(3).  
  MOVE 4 TO VLTHS(1).  
  MOVE 4 TO VLTHS(2).  
  MOVE 4 TO VLTHS(3).  
  MOVE 20 TO VVALS(1).  
  MOVE 40 TO VVALS(2).  
  MOVE 84 TO VVALS(3).  
  MOVE 3 TO QIPNUM.  
  CALL DSQCIB USING DSQCOMM, QICLTH, SETG,  
                  QIPNUM, QIKLTHS, VNAMES,  
                  QIVLTHS, VVALUES, DSQ-VARIABLE-FINT.  
*  
* Run a Query  
  MOVE 12 TO QICLTH.  
  CALL DSQCIB USING DSQCOMM, QICLTH, QUERY.  
*  
* Print the results of the query  
  MOVE 22 TO QICLTH.  
  CALL DSQCIB USING DSQCOMM, QICLTH, REPT.  
*  
* End the query interface session  
  MOVE 4 TO QICLTH.  
  CALL DSQCIB USING DSQCOMM, QICLTH, ENDQI.  
  
STOP RUN.
```

그림 43. 샘플 COBOL 프로그램(DSQABFCO) (2/2)

CICS의 경우, STOP RUN 문은 GOBACK 문으로 변경되어야 합니다.

COBOL용 DSQCOMM

이 포함 파일은 DSQCOMMB이라 불리며 QMF와 함께 제공됩니다.


```

*****
* COBOL INCLUDE FOR QUERY CALLABLE INTERFACE (MVS/VM)
*****
00001000
00002000
00003000
00004000
* STRUCTURE DECLARE FOR COMMUNICATIONS AREA
00005000
00006000
01 DSQCOMM.
00007000
00008000
03 DSQ-RETURN-CODE PIC 9(8) USAGE IS COMP.
00009000
* FUNCTION RETURN CODE *
00010000
* 03 DSQ-INSTANCE-ID PIC 9(8) USAGE IS COMP.
00011000
* IDENTIFIER FROM START CMD *
00012000
* 03 DSQ-COMM-LEVEL PIC X(12).
00013000
* COMMUNICATIONS LEVEL *
00014000
* 03 DSQ-PRODUCT PIC X(2).
00015000
* QUERY PRODUCT ID *
00016000
* 03 DSQ-PRODUCT-RELEASE PIC X(2).
00017000
* QUERY PRODUCT RELEASE *
00018000
* 03 DSQ-RESERVE1 PIC X(28).
00019000
* RESERVED AREA *
00020000
* 03 DSQ-MESSAGE-ID PIC X(8).
00021000
* COMPLETION MESSAGE ID *
00022000
* 03 DSQ-Q-MESSAGE-ID PIC X(8).
00023000
* QUERY MESSAGE ID *
00024000
* 03 DSQ-START-PARM-ERROR PIC X(8).
00025000
* START PARAMETER IN ERROR *
00026000
* 03 DSQ-CANCEL-IND PIC X(1).
00027000
* 1 = COMMAND CANCELLED *
00028000
* 0 = COMMAND NOT CANCELLED *
00029000
* 03 DSQ-RESERVE2 PIC X(23).
00030000
* RESERVED AREA *
00031000
* 03 DSQ-RESERVE3 PIC X(156).
00032000
* RESERVED AREA *
00033000
* 03 DSQ-MESSAGE-TEXT PIC X(128).
00034000
* QMF MESSAGE TEXT *
00035000
* 03 DSQ-Q-MESSAGE-TEXT PIC X(128).
00036000
* QMF QUERY MESSAGE TEXT *
00037000
* 512 BYTES TOTAL *
00038000
00039000
* VALUES FOR DSQ-RETURN-CODE
00040000
00041000
00042000
01 DSQ-SUCCESS PIC 9(8) USAGE IS COMP VALUE 0.
00043000
01 DSQ-WARNING PIC 9(8) USAGE IS COMP VALUE 4.
00044000
01 DSQ-FAILURE PIC 9(8) USAGE IS COMP VALUE 8.
00045000
01 DSQ-SEVERE PIC 9(8) USAGE IS COMP VALUE 16.
00046000
00047000
* VALUES FOR DSQ-INSTANCE-ID
00048000
00049000
01 DSQ-CONTINUE PIC 9(8) USAGE IS COMP VALUE 0.
00050000

```

그림 44. COBOL 통신 영역(DSQCOMMB) (1/2)

```

* VALUES FOR DSQ-COMM-LEVEL
01 DSQ-CURRENT-COMM-LEVEL PIC X(12) VALUE "DSQL>001002<".
* VALUES FOR DSQ-PRODUCT
01 DSQ-QRW          PIC X(2) VALUE "01".
01 DSQ-QMF          PIC X(2) VALUE "02".
01 DSQ-QM4          PIC X(2) VALUE "03".
* VALUES FOR DSQ-PRODUCT-RELEASE
01 DSQ-QRW-V1R2     PIC X(2) VALUE "01".
01 DSQ-QRW-V1R3     PIC X(2) VALUE "02".
01 DSQ-QMF-V2R4     PIC X(2) VALUE "01".
01 DSQ-QMF-V3R1     PIC X(2) VALUE "02".
01 DSQ-QMF-V3R1M1   PIC X(2) VALUE "03".
01 DSQ-QMF-V3R2     PIC X(2) VALUE "04".
01 DSQ-QMF-V3R3     PIC X(2) VALUE "05".
01 DSQ-QMF-V6R1     PIC X(2) VALUE "06".
01 DSQ-QM4-V1R1     PIC X(2) VALUE "01".
* VALUES FOR DSQ-CANCEL-INDE
01 DSQ-CANCEL-YES   PIC X(1) VALUE "1".
01 DSQ-CANCEL-NO    PIC X(1) VALUE "0".
* VALUES FOR MODE
01 DSQ-INTERACTIVE  PIC X(1) VALUE "1".
01 DSQ-BATCH        PIC X(1) VALUE "2".
* VALUES YES AND NO
01 DSQ-YES          PIC X(1) VALUE "1".
01 DSQ-NO           PIC X(1) VALUE "2".
* CALLABLE INTERFACE PROGRAM NAME
01 DSQCIB           PIC X(6) VALUE "DSQCIB".
* VALUES FOR VARIABLE TYPE ON CALL PARAMETER
01 DSQ-VARIABLE-CHAR PIC X(4) VALUE "CHAR".
01 DSQ-VARIABLE-FINT PIC X(4) VALUE "FINT".

```

그림 44. COBOL 통신 영역(DSQCOMMB) (2/2)

COBOL 호출 인터페이스 프로그램 실행에 대한 고려사항

QMF 호출 인터페이스를 사용하는 프로그램을 변환하고, 컴파일하고, 링크 편집할 경우 다음을 고려하십시오.

- 실행 환경

QMF는 COBOL 환경에서 어셈블러 서브프로그램으로 실행됩니다. COBOL 프로그램은 COBOL 동적 호출을 사용하여 QMF 인터페이스 프로그램 DSQCIB를 호출해야 합니다.

- 큰 따옴표 또는 작은 따옴표

COBOL 프로그램에서 리터럴을 분리하려면 큰 따옴표(")나 작은 따옴표(') 중 하나를 사용하십시오. 선택한 구분문자를 QUOTE 또는 APOST를 지정하여 CICS 변환 프로세스 또는 COBOL 컴파일러에 지정할 수 있습니다. 실제로 COBOL 컴파일러의 APOST 또는 QUOTE 옵션이 CICS 변환기의 옵션과 일치하는지 확인하십시오.

QMF에 의해 분산된 통신 영역 DSQCOMMB 및 샘플 COBOL 프로그램 DSQABFCO는 큰 따옴표를 사용하여 리터럴을 분리합니다. 대신 설치 과정 또는 프로그램에서 작은 따옴표가 사용될 경우 QMF에 의해 분산된 것으로서 DSQCOMMB을 변경하거나 프로그램에 구조를 복사하여 큰 따옴표를 작은 따옴표로 바꾸십시오.

- 통신 매크로 DSQCOMMB

통신 영역 DSQCOMMB를 COBOL 컴파일 단계에 사용해야 하며 혹은 제어 구조로서 프로그램에 복사해야 합니다.

- 인터페이스 모듈 DSQCIB

프로그램의 링크 편집 단계 동안 QMF 인터페이스 모듈을 사용할 수 있어야 합니다.

CICS에서 COBOL 프로그램 실행

프로그램을 작성한 후에는 실행하기 전에 필요에 따라 프로그램을 변환하고, 컴파일하고, 링크 편집해야 합니다. 이 절에 나열된 프로그램에서는 반드시 수행해야 할 단계를 보여줍니다.

QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

COBOL 언어 인터페이스

MVS에서 CICS의 변환, 컴파일, 링크 편집

다음 예제에서는 CICS 제공 프로시저 DFHEBTVL을 보여줍니다. 이는 COBOL을 지원합니다. CICS에서 사용할 수 있도록 프로그램을 변환하는 방법에 관한 자세한 내용은 CICS 라이브러리를 참조하십시오.

```
//samCOBOL JOB
//          EXEC PROC=DFHEBTVL
//TRN.SYSIN DD *
*CBL      XOPTS(CICS translator options ...QUOTE COBOL2)
          .
          Your program or copy of QMF sample DSQBFCO
          .
/*
/** Provide Access to QMF Communications Macro DSQCOMMB
//COB.SYSLIB DD DSN=QMF710.SDSQSAPE,DISP=SHR
/** Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF710.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
          INCLUDE CICSLOAD(DFHECI)
          INCLUDE QMFLOAD(DSQCIB)
          ORDER DFHECI
          ENTRY samCOBOL
          MODE AMODE(31) RMODE(ANY)
          NAME samCOBOL(R)
/*
```

그림 45. CICS 변환기, COBOL 컴파일러 및 연계 편집기를 실행하는 JCL

VSE에서 CICS의 변환, 컴파일 및 링크 편집

201 페이지의 그림 46에서 VSE 작업 제어는 VSE에서 실행되는 CICS에 COBOL 프로그램을 설치하는 예제입니다. COBOL 프로그램을 변환하고 컴파일하는 방법에 관한 자세한 내용은 CICS 라이브러리를 참조하십시오.

QMF에 제공된 이 예제는 QMF 서브라이브러리에 위치하며 이름은 DSQ3CICO.Z입니다.

```

* $$ JOB JNM=DSQ3CICO,DISP=D,CLASS=0
// JOB DSQ3CICO   Sample job to Install QMF Callable Interface (COBOL)
* -----
* Install QMF Callable Interface Example (COBOL)
* -----
// SETPARM VOLID=valid      *-- update valid for syspch
// SETPARM START=rtrk      *-- update start track/block (syspch)
// SETPARM SIZE=ntrks      *-- update number of tracks/blocks (syspch)
* -----
// DLBL   IJSYSYPH,'CICS.TRANSLAT.OUTPUT',0
// EXTENT SYSPCH,,1,0,&START,&SIZE
ASSGN SYSPCH,DISK,VOL=&VOLID,SHR
* Library search chain must contain the QMF, CICS and COBOL sublibrary
// LIBDEF *,SEARCH=(PRD2.PROD,PRD1.BASE,PRD2.CONFIG)
// LIBDEF PHASE,CATALOG=PRD2.PROD
* -----
* Step 1: Translate callable interface program (COBOL)
* -----
* You may need to update or remove the SLI statement for your program.
* -----
// EXEC   DFHECP1$,SIZE=256K,PARM='XOPTS(CICS,QUOTE)'
* $$ SLI MEM=DSQABFCO.Z,S=PRD2.PROD
/*
* -----
* Step 2: Compile callable interface program (COBOL)
* -----
CLOSE SYSPCH,000
// DLBL   IJSYSIN,'CICS.TRANSLAT.OUTPUT',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLID,SHR
// OPTION NODECK,CATAL
      PHASE DSQABFCO,*
      INCLUDE DFHECI
// EXEC IGYCRCTL,PARM='SZ(MAX),OBJECT,MAP,RES,NODYNAM,QUOTE,LIB,RENT'
CLOSE SYSIPT,SYSRDR
/*
* -----
* Step 3: Link-edit callable interface program (COBOL)
* -----
// EXEC LNKEDT,PARM='AMODE=31,RMODE=ANY'
/*
/&
// JOB RESET
ASSGN SYSIPT,SYSRDR   IF 1A93D, CLOSE SYSIPT,SYSRDR
ASSGN SYSPCH,000     IF 1A93D, CLOSE SYSPCH,000
/&
* $$ E0J

```

그림 46. CICS/VSE 변환기, COBOL 컴파일러 및 연계 편집기를 실행하는 작업 제어

VM의 CMS에서 프로그램 컴파일 및 실행

다음 프로그램은 IBM COBOL 컴파일러를 사용하여 호출 인터페이스 응용프로그램을 컴파일하고 실행합니다.

QMF는 이 예제에서 REXX EXEC를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

```
/* **** */
/* Compile your COBOL program and run it.          */
/* **** */
TRACE off
ADDRESS CMS
/* **** */
/* Access COBOL product disk using a program, PRODUCT, that you */
/* write.                                           */
/* **** */
"EXEC PRODUCT COBOL"
/* **** */
/* Get QMF DSQCOMM into a macro library and set GLOBAL compile */
/* maclibs.                                        */
/* **** */
"ERASE TEMPP MACLIB A"
"MACLIB GEN TEMPP DSQCOMMB"
MacList = "TEMPP VSC2MAC COB2MLIB COB2PLIB DMSSP CMSLIB OSMACRO"
"GLOBAL MACLIB" MacList
/* **** */
/* Compile the program                            */
/* **** */
"GLOBAL TXTLIB SCEELKED"
"COBOL2 yourname (LIB RESIDENT LIST RENT DYNAM"
```

그림 47. CMS에서 COBOL을 컴파일하고 실행하는 프로그램 (1/2)

```

/*****/
/* Access SQL/DS and initialize database */
/*****/
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"
/*****/
/* Access GDDM product disk */
/*****/
"EXEC PRODUCT GDDM"
/*****/
/* Issue Filedefs for QMF product */
/*****/
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A ( PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 ( PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * ( PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQLDLIB LOADLIB *"
/*****/
/* Provide access to QMF and COBOL program libraries */
/*****/
"GLOBAL LOADLIB DSQLDLIB VSC2LOAD"
"GLOBAL TXTLIB VSC2LTXT ADMRLIB ADMPLIB ADMGLIB SCEELKED"
Say "Starting to run COBOL program"
"RUN yourname"
Exit 0

```

그림 47. CMS에서 COBOL을 컴파일하고 실행하는 프로그램 (2/2)

TSO에서 COBOL 프로그램 실행

다음 절은 TSO(ISPF 유무에 관계없음)에서 컴파일된 프로그램을 실행하기 위해 COBOL 컴파일러, 연계 편집기 및 샘플 프로그램을 실행하기 위한 샘플 JCL을 제공합니다.

COBOL 언어 인터페이스

TSO에서 컴파일 및 링크 편집

다음 작업은 COBOL 컴파일러를 사용하여 호출 인터페이스 응용프로그램을 컴파일합니다. 그런 다음, 응용프로그램을 링크 편집합니다. 일부 매개변수의 경우 설치에 따라 달라질 수 있습니다. 자세한 내용은 QMF 관리자에게 문의하십시오.

```
//samCOBOL JOB
//STEP1 EXEC PROC=IGYWCL
//* Provide Access to QMF Communications Macro DSQCOMM
//COBOL.SYSLIB DD DSN=QMF710.SAMPLIB,DISP=SHR
//COBOL.SYSIN DD *
        .
        Your program or copy of QMF sample DSQABFCO
        .
/*
/** Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF710.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
        INCLUDE QMFLOAD(DSQCIB)
        ENTRY samCOBOL
        MODE AMODE(31) RMODE(ANY)
        NAME samCOBOL(R)
/*
```

그림 48. COBOL 컴파일러 및 연계 편집기를 실행하는 JCL

ISPF 없이 TSO에서 프로그램 실행

프로그램을 컴파일한 후에는 다음과 유사한 프로그램을 작성하여 이를 수행할 수 있습니다.


```

PROC 0
CONTROL ASIS
/*****/
/* Note: QMF, DB2, GDDM and COBOL load libraries must be */
/* allocated before running this CLIST. */
/* Name of QMF load library is "QMF710.SDSQLOAD". */
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF710.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF710.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command */
/*****/
CALL samCOBOL
EXIT CODE(0)

```

그림 49. COBOL 컴파일러 및 연계 편집기를 실행하는 JCL

ISPF하의 TSO에서 프로그램 실행

프로그램을 컴파일한 후에는 다음과 유사한 프로그램을 작성하여 이를 수행할 수 있습니다.

```

PROC 0
CONTROL ASIS
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF710.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD', +
    'PRODUCT.COB2LIB')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF710.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF710.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF710.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF710.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF710.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(samCOBOL) NEWAPPL(DSQE)
EXIT CODE(4)

```

그림 50. ISPF하의 TSO에서 프로그램을 실행하는 CLIST

EXIT CODE(4)는 ISPF 특성 패널이 표시되지 않도록 합니다.

FORTRAN 언어 인터페이스

FORTRAN 호출 인터페이스는 다른 SAA 언어에 대해 제공된 인터페이스와 일치합니다.

CICS 사용자를 위한 주의사항

FORTRAN은 CICS에서는 사용할 수 없으므로 FORTRAN용 QMF 호출 인터페이스는 CICS에서 작동하지 않습니다.

FORTRAN의 인터페이스 통신 영역 맵핑(DSQCOMMF)

DSQCOMMF는 FORTRAN에 대한 DSQCOMM 맵핑을 제공하고 해당 제품과 함께 제공됩니다. 표18에서는 수정하면 안되는 DSQCOMMF에 대한 정보를 보여줍니다.

표 18. 인터페이스 통신 영역(DSQCOMMF)

구조 이름	데이터 유형	설명
DSQ_RETURN_CODE	INTEGER	QMF 명령이 실행된 후의 상태를 나타냅니다. 해당 값은 다음과 같습니다. DSQ_SUCCESS 요청의 성공적인 실행 DSQ_WARNING 경고 메시지를 나타내는 정상적인 완료 DSQ_FAILURE 명령이 제대로 실행되지 않음 DSQ_SEVERE 심각한 오류. QMF 세션이 종료됨
DSQ_INSTANCE_ID	INTEGER	START 명령의 실행 동안 QMF에 의해 설정된 ID
DSQ_COMM_LEVEL	CHARACTER(12)	DSQCOMM의 레벨을 식별합니다. 이것을 DSQ_CURRENT_COMM_LEVEL 값으로 설정한 다음 QMF START 명령을 발행합니다.
DSQ_PRODUCT	CHARACTER(2)	사용중인 IBM 조회 제품을 식별합니다.

FORTRAN 언어 인터페이스

표 18. 인터페이스 통신 영역(DSQCOMMF) (계속)

구조 이름	데이터 유형	설명
DSQ_PRODUCT_RELEASE	CHARACTER(2)	사용중인 조회 제품의 릴리스 레벨을 식별합니다.
DSQ_RESERVE1	CHARACTER(28)	앞으로 사용되도록 예약됨
DSQ_MESSAGE_ID	CHARACTER(8)	완료 메시지 ID
DSQ_Q_MESSAGE_ID	CHARACTER(8)	조회 메시지 ID
DSQ_START_PARM_ERROR	CHARACTER(8)	START가 매개변수 오류로 인해 실패했을 경우 오류 상태에 있는 매개변수
DSQ_CANCEL_IND	CHARACTER(1)	QMF 명령이 실행되는 동안 사용자가 취소했을 경우 다음 두 값 중 하나를 포함합니다. DSQ_CANCEL_YES CHARACTER(1) DSQ_CANCEL_NO CHARACTER(1)
DSQ_RESERVE2	CHARACTER(23)	앞으로 사용되도록 예약됨
DSQ_RESERVE3	CHARACTER(156)	앞으로 사용되도록 예약됨
DSQ_MESSAGE_TEXT	CHARACTER(128)	완료 메시지 텍스트
DSQ_Q_MESSAGE_TEXT	CHARACTER(128)	조회 메시지 텍스트

FORTRAN의 함수 호출

QMF는 FORTRAN 언어에 대해 두 개의 함수 호출인 DSQCIF 및 DSQCIFE를 제공합니다. 두 가지 호출에 대해서는 통신 매크로 DSQCOMMF에서 설명됩니다.

DSQCIF

이 호출은 응용프로그램 변수에 액세스를 요구하지 않는 QMF 명령에 대한 것입니다. 대부분의 QMF 명령에서 이 호출을 사용하십시오.

```
RC = DSQCIF(DSQCOMM,
+   CMDLTH,
+   CMDSTR)
```

매개변수 값은 다음과 같습니다.

DSQCOMM

통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이. 이 값은 정수 매개변수입니다.

CMDSTR

실행할 QMF 명령. 이 값은 CMDLTH에 의해 지정된 길이의 대문자 문자열입니다.

DSQCIFE

이 호출은 응용프로그램 변수에 액세스를 요구하는 세 가지 명령에 대한 확장 구문을 보유합니다. 세 명령은 START와 GET GLOBAL 및 SET GLOBAL의 확장 형식입니다.

이 호출에 대한 구문은 다음과 같습니다.

```

RC = DSQCIFE(DSQCOMM,
+   CMDLTH,
+   CMDSTR,
+   PNUM,
+   KLTH,
+   KWORD,
+   VLTH,
+   VALUE,
+   VTYPE)

```

매개변수 값은 다음과 같습니다.

DSQCOMM

인터페이스 통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이. 이 값은 정수 매개변수입니다.

CMDSTR

실행할 QMF 명령. 이 값은 CMDLTH에 의해 지정된 길이의 대문자 문자열입니다.

PNUM

명령 키워드의 수. 이 값은 정수 매개변수입니다.

KLTH

각 지정된 키워드의 길이. 이 값은 정수 매개변수 또는 매개변수 배열입니다.

KEYWORD

QMF 키워드. 이 값은 KLTH에 의해 지정된 것과 같은 길이를 가진 문자 또는 문자의 구조입니다. 모든 키워드가 같은 길이이면 문자의 배열을 사용할 수 있습니다. QMF는 키워드가 연속적인 기억영역에 있고 어떤 특수 분리 문자에 의해 분리되지 않는다고 가정합니다.

VLTH

키워드와 연관된 각 값의 길이는 정수 매개변수 또는 매개변수 배열입니다.

VALUE

각 키워드와 연관된 값. 유형은 VTYPE 매개변수에 지정되어 있으며 문자, 문자의 구조, 정수 매개변수 또는 매개변수 배열이 될 수 있습니다. 문자값이 있으면 QMF는 값이 연속적인 기억영역에 있고 어떤 특수 분리 문자에 의해 분리되지 않는다고 가정합니다.

VTYPE

값 문자열 VALUE의 QMF 데이터 유형. VTYPE는 두 값 중에 하나를 가질 수 있으며 이 값은 통신 매크로 DSQCOMMVF에서 제공됩니다.

문자 값의 경우 DSQ_VARIABLE_CHAR.

정수 값의 경우 DSQ_VARIABLE_FINT.

VALUE 필드에서 지정된 모든 값은 VTYPE에서 지정된 데이터 유형을 가집니다.

FORTRAN 프로그래밍 예제

다음 프로그램 DSQABFF는 QMF와 함께 제공되고 VS FORTRAN를 사용합니다.

여기에서 샘플 소스 코드 목록을 볼 수 있고 온라인으로 액세스할 수도 있습니다. OS/390의 경우, 샘플 프로그램은 라이브러리 QMF710.SDSQSAPE의 멤버입니다. VM의 경우, 샘플 프로그램은 제품 디스크에 있습니다.

FORTRAN 호출 인터페이스의 샘플 프로그램은 다음 기능을 수행합니다.

- QMF를 시작합니다.
- 세 개의 글로벌 변수를 설정합니다.
- Q1이라는 조회를 실행합니다.

- 서식 F1을 사용하여 결과 보고서를 인쇄합니다.
- QMF 세션을 종료합니다.

QMF는 조회 Q1 또는 서식 F1을 제공하지 않으나, 샘플 프로그램은 이 오브젝트를 사용합니다.

이 절에서는 호출 인터페이스를 사용하여 FORTRAN 프로그램을 컴파일하고, 링크 편집하고, 실행하는 방법을 보여줍니다. QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

```

C*****
C Sample Program: dsqabff
C FORTRAN Version of SAA Query Manager Callable Interface
CC Creation Date: 11/21/89
CC ENVIRONMENT:   API IN FORTRAN
C*****
CC Processing:
C     a. Start a Query Manager Session using the Callable Interface.
CC     b. Set Global Query Manager numeric variables.
CC     d. Run a Query Manager query using the Callable Interface.
CC     e. Print a report using the Callable Interface.
CC     f. Exit the Query Manager Session.
C
    
```

그림 51. 샘플 FORTRAN 프로그램(DSQABFF) (1/5)

FORTRAN 언어 인터페이스

```
C Prerequisites:1. Create the SAMPLE database.
CC                2. Create a prompted query, Q1, which has a SELECT state
CC                3. Create a form, F1, that displays data for query Q1.
CC*****
```

```
PROGRAM DSQABFF
```

```
C*****
C Include and declare query interface communications area
C*****
INCLUDE (DSQCOMMF)

C*****
C Query interface command lengths and commands
C*****
INTEGER COMMAND_LENGTH
CHARACTER START_QUERY INTERFACE*5,
+         SET_GLOBAL_VARIABLES*10,
+         RUN_QUERY*12,
+         PRINT_REPORT*22,
+         END_QUERY_INTERFACE*4

C*****
C Query command extension, number of parameters and lengths
C*****
INTEGER NUMBER_OF_PARAMETERS,
+         KEYWORD_LENGTHS(10),
+         DATA_LENGTHS(10)

C*****
C Variable data type constants
C*****
CHARACTER CHAR_DATA_TYPE*4,
+         INT_DATA_TYPE*4

C*****
C Keyword parameter and value for START command
C*****
CHARACTER*8 START_KEYWORDS(1)
CHARACTER*11 START_KEYWORD_VALUES(1)
```

그림 51. 샘플 FORTRAN 프로그램(DSQABFF) (2/5)


```

C*****
C Keyword parameter and values for SET command
C*****
CHARACTER SET_KEYWORDS(19)
CHARACTER SET_KEYWORD_1*7,
+ SET_KEYWORD_2*5,
+ SET_KEYWORD_3*7

EQUIVALENCE (SET_KEYWORDS( 1), SET_KEYWORD_1),
+ (SET_KEYWORDS( 8), SET_KEYWORD_2),
+ (SET_KEYWORDS(13), SET_KEYWORD_3)

CHARACTER SET_VALUES(12)
INTEGER*4 SET_VALUE_1,
+ SET_VALUE_2,
+ SET_VALUE_3

EQUIVALENCE (SET_VALUES(1), SET_VALUE_1),
+ (SET_VALUES(5), SET_VALUE_2),
+ (SET_VALUES(9), SET_VALUE_3)

C*****
C Declare command length and return code variables
C*****
INTEGER LEN,
+ RC

C*****
C Initialization
C*****

DATA START_QUERY_INTERFACE /'START' /
DATA SET_GLOBAL_VARIABLES /'SET GLOBAL' /
DATA RUN_QUERY /'RUN QUERY Q1' /
DATA PRINT_REPORT /'PRINT REPORT (FORM=F1)'/
DATA END_QUERY_INTERFACE /'EXIT' /

DATA CHAR_DATA_TYPE /DSQ_VARIABLE_CHAR /
DATA INT_DATA_TYPE /DSQ_VARIABLE_FINT /

```

그림 51. 샘플 FORTRAN 프로그램(DSQABFF) (3/5)

FORTRAN 언어 인터페이스

```
C*****
C  Start Query Session
C*****

      DSQ_COMM_LEVEL = DSQ_CURRENT_COMM_LEVEL
      NUMBER_OF_PARAMETERS = 1
      COMMAND_LENGTH = LEN(START_QUERY_INTERFACE)
      KEYWORD_LENGTHS(1) = LEN(START_KEYWORDS(1))
      DATA_LENGTHS(1) = LEN(START_KEYWORD_VALUES(1))
      START_KEYWORDS(1) = 'DSQSMODET'
      START_KEYWORD_VALUES(1) = 'INTERACTIVE'

      RC = DSQCIFE(DSQCOMM,
+               COMMAND_LENGTH,
+               START_QUERY_INTERFACE,
+               NUMBER_OF_PARAMETERS,
+               KEYWORD_LENGTHS,
+               START_KEYWORDS,
+               DATA_LENGTHS,
+               START_KEYWORD_VALUES,
+               CHAR_DATA_TYPE)

C*****
C  Set numeric values into query using SET command
C*****

      NUMBER_OF_PARAMETERS = 3
      COMMAND_LENGTH = LEN(SET_GLOBAL_VARIABLES)
      SET_KEYWORD_1 = 'MYVAR01'
      SET_KEYWORD_2 = 'SHORT'
      SET_KEYWORD_3 = 'MYVAR03'
      KEYWORD_LENGTHS(1) = LEN(SET_KEYWORD_1)
      KEYWORD_LENGTHS(2) = LEN(SET_KEYWORD_2)
      KEYWORD_LENGTHS(3) = LEN(SET_KEYWORD_3)
      DATA_LENGTHS(1) = 4
      DATA_LENGTHS(2) = 4
      DATA_LENGTHS(3) = 4
      SET_VALUE_1 = 20
      SET_VALUE_2 = 40
      SET_VALUE_3 = 84

      RC = DSQCIFE(DSQCOMM,
+               COMMAND_LENGTH,
+               SET_GLOBAL_VARIABLES,
+               NUMBER_OF_PARAMETERS,
+               KEYWORD_LENGTHS,
+               SET_KEYWORDS,
+               DATA_LENGTHS,
+               SET_VALUES,
+               INT_DATA_TYPE)
```

그림 51. 샘플 FORTRAN 프로그램(DSQABFF) (4/5)

```

C*****
C  Run a query
C*****
      COMMAND_LENGTH = LEN(RUN_QUERY)
      RC = DSQCIF(DSQCOMM,
+               COMMAND_LENGTH,
+               RUN_QUERY)

C*****
C  Print the results of the query
C*****
      COMMAND_LENGTH = LEN(PRINT_REPORT)
      RC = DSQCIF(DSQCOMM,
+               COMMAND_LENGTH,
+               PRINT_REPORT)

C*****
C  End the query interface session
C*****
      COMMAND_LENGTH = LEN(END_QUERY_INTERFACE)
      RC = DSQCIF(DSQCOMM,
+               COMMAND_LENGTH,
+               END_QUERY_INTERFACE)

      END
    
```

그림 51. 샘플 FORTRAN 프로그램(DSQABFF) (5/5)

FORTRAN용 DSQCOMM

이 파일은 DSQCOMM.F라 불리며 QMF와 함께 제공됩니다.

FORTRAN 언어 인터페이스

```

C*****00001000
C      FORTRAN include file for Callable Interface (MVS/VM)00002000
C*****00003000
C      Return codes00004000
      INTEGER DSQ_SUCCESS, DSQ_WARNING, DSQ_FAILURE, DSQ_SEVERE00005000
      PARAMETER(00006000
+         DSQ_SUCCESS = 0,00007000
+         DSQ_WARNING = 4,00008000
+         DSQ_FAILURE = 8,00009000
+         DSQ_SEVERE = 16)00010000
00011000
C      Communications level00012000
      CHARACTER DSQ_CURRENT_COMM_LEVEL*1200013000
      PARAMETER(00014000
+         DSQ_CURRENT_COMM_LEVEL = 'DSQL>001002<')00015000
00016000
C      Query product IDs00017000
      CHARACTER DSQ_QRW*2, DSQ_QMF*2, DSQ_QM4*200018000
      PARAMETER(00019000
+         DSQ_QRW = '01',00020000
+         DSQ_QMF = '02',00021000
+         DSQ_QM4 = '03')00022000
00023000
C      Query product release levels00024000
      CHARACTER DSQ_QRW_V1R2*2, DSQ_QRW_V1R3*2,00025000
+         DSQ_QMF_V2R4*2, DSQ_QMF_V3R1*2,00026000
+         DSQ_QMF_V3R1M1*2, DSQ_QMF_V3R2*2,00027000
+         DSQ_QMF_V3R3*2, DSQ_QMF_V6R1*2,00028000
+         DSQ_QM4_V1R1*200029000
      PARAMETER(00030000
+         DSQ_QRW_V1R2 = '01',00031000
+         DSQ_QRW_V1R3 = '02',00032000
+         DSQ_QMF_V2R4 = '01',00033000
+         DSQ_QMF_V3R1 = '02',00034000
+         DSQ_QMF_V3R1M1 = '03',00035000
+         DSQ_QMF_V3R2 = '04',00036000
+         DSQ_QMF_V3R3 = '05',00037000
+         DSQ_QMF_V6R1 = '06',00038000
+         DSQ_QM4_V1R1 = '01')00039000
00040000
C      Host variable types00041000
      CHARACTER DSQ_VARIABLE_CHAR*4, DSQ_VARIABLE_FINT*400042000
      PARAMETER(00043000
+         DSQ_VARIABLE_CHAR = 'CHAR',00044000
+         DSQ_VARIABLE_FINT = 'FINT')00045000
00046000
C      Cancel indicator00047000
      CHARACTER DSQ_CANCEL_YES, DSQ_CANCEL_NO00048000
      PARAMETER(00049000
+         DSQ_CANCEL_YES = '1',00050000
+         DSQ_CANCEL_NO = '0')00051000
00052000
      CHARACTER DSQCOMM(512)00053000

```

그림 52. FORTRAN 통신 영역(DSQCOMMF) (1/2)

INTEGER	DSQ_RETURN_CODE, DSQ_INSTANCE_ID	00054000
CHARACTER	DSQ_COMM_LEVEL*12,	00055000
+	DSQ_PRODUCT*2,	00056000
+	DSQ_PRODUCT_RELEASE*2,	00057000
+	DSQ_RESERVE1*28,	00058000
+	DSQ_MESSAGE_ID*8,	00059000
+	DSQ_Q_MESSAGE_ID*8,	00060000
+	DSQ_START_PARM_ERROR*8,	00061000
+	DSQ_CANCEL_IND*1,	00062000
+	DSQ_RESERVE2*23,	00063000
+	DSQ_RESERVE3*156,	00064000
+	DSQ_MESSAGE_TEXT*128,	00065000
+	DSQ_Q_MESSAGE_TEXT*128	00066000
		00067000
EQUIVALENCE	(DSQCOMM(1), DSQ_RETURN_CODE),	00068000
+	(DSQCOMM(5), DSQ_INSTANCE_ID),	00069000
+	(DSQCOMM(9), DSQ_COMM_LEVEL),	00070000
+	(DSQCOMM(21), DSQ_PRODUCT),	00071000
+	(DSQCOMM(23), DSQ_PRODUCT_RELEASE),	00072000
+	(DSQCOMM(25), DSQ_RESERVE1),	00073000
+	(DSQCOMM(53), DSQ_MESSAGE_ID),	00074000
+	(DSQCOMM(61), DSQ_Q_MESSAGE_ID),	00075000
+	(DSQCOMM(69), DSQ_START_PARM_ERROR),	00076000
+	(DSQCOMM(77), DSQ_CANCEL_IND),	00077000
+	(DSQCOMM(78), DSQ_RESERVE2),	00078000
+	(DSQCOMM(101), DSQ_RESERVE3),	00079000
+	(DSQCOMM(257), DSQ_MESSAGE_TEXT),	00080000
+	(DSQCOMM(385), DSQ_Q_MESSAGE_TEXT)	00081000
		00082000
C	Callable Interface Normal and Extended Calls	00083000
	EXTERNAL DSQCIF	00084000
	EXTERNAL DSQCIFE	00085000

그림 52. FORTRAN 통신 영역(DSQCOMM) (2/2)

VM의 CMS에서 프로그램 컴파일 및 실행

다음 프로그램은 VS FORTRAN 컴파일러를 사용하여 호출 인터페이스 응용프로그램을 컴파일하고 실행합니다. QMF는 이 예제에서 REXX EXEC를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

FORTRAN 언어 인터페이스

```

/*****/
/* Compile your program and run it. */
/*****/
TRACE off
ADDRESS CMS

/*****/
/* Access FORTRAN product disk using a program, PRODUCT, that you */
/* write. */
/*****/
"EXEC PRODUCT FORTRAN"

/*****/
/* Get QMF DSQCOMM into a macro library and set GLOBAL compile */
/* maclibs. */
/*****/
"ERASE TEMPP MACLIB A"
"MACLIB GEN TEMPP DSQCOMM"
MacList = "TEMPP VSF2PLIB VSF2MLIB DMSSP CMSLIB OSMACRO"
"GLOBAL MACLIB" MacList

/*****/
/* Compile the program */
/*****/
'FORTVS2 yourname (RENT OPT(0) XREF'

/*****/
/* Access SQL/DS and initialize database */
/*****/
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"

/*****/
/* Access GDDM product disk */
/*****/
"EXEC PRODUCT GDDM"

```

그림 53. 프로그램을 컴파일하고 실행하는 REXX 프로그램 (1/2)

```

/*****/
/* Issue Filedefs for QMF product */
/*****/
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A ( PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 ( PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * ( PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQDLIB LOADLIB *"

/*****/
/* Provide access to QMF and FORTRAN program libraries */
/*****/
'GLOBAL LOADLIB VSF2LOAD DSQDLIB'
'GLOBAL TXTLIB VSF2LINK VSF2FORT ADMRLIB ADMPLIB ADMGLIB'
Say "Starting to run FORTRAN program"
"RUN yourname"

Exit 0

```

그림 53. 프로그램을 컴파일하고 실행하는 REXX 프로그램 (2/2)

설치에 맞게 이 프로그램을 변경해야 합니다.

MVS의 TSO에서 프로그램 실행

프로그램을 작성한 후에는 실행하기 전에 필요에 따라 프로그램을 컴파일하고, 링크 편집해야 합니다. 이 단원에서는 나열된 프로그램은 반드시 수행해야 할 단계를 보여줍니다.

QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

FORTRAN 언어 인터페이스

TSO에서 컴파일 및 링크 편집

다음 작업은 MVS용 VS FORTRAN 컴파일러를 사용하여 호출 인터페이스 응용프로그램을 컴파일하고 링크 편집합니다. 일부 매개변수는 설치에 따라 달라질 수 있습니다. 자세한 내용은 QMF 관리자에게 문의하십시오.

```
//samFORT    JOB
//STEP1      EXEC PROC=VSF2CL
/** Provide Access to QMF Communications Macro DSQCOMM
//FORT.SYSLIB DD DSN=QMF710.SAMPLIB,DISP=SHR
//FORT.SYSIN  DD *
                .
                Your program or copy of QMF sample DSQABFF
                .
/*
/** Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF710.SDSQLOAD,DISP=SHR
//LKED.SYSIN   DD *
                INCLUDE QMFLOAD(DSQCIF)
                INCLUDE QMFLOAD(DSQCIF)
                ENTRY samFORT
                MODE  AMODE(31) RMODE(ANY)
                NAME  samFORT(R)
/*
```

그림 54. FORTRAN 컴파일러 및 연계 편집기를 실행하는 JCL

ISPF 없이 TSO에서 프로그램 실행

다음 프로그램은 VS FORTRAN 컴파일러를 사용하여 호출 인터페이스 응용프로그램을 실행합니다. 일부 매개변수는 설치에 따라 달라질 수 있습니다. 자세한 내용은 QMF 관리자에게 문의하십시오.


```

PROC 0
CONTROL ASIS
/*****/
/* Note: QMF, DB2, GDDM and FORTRAN load libraries must be */
/*     allocated before running this CLIST.                */
/*     Name of QMF load library is "QMF710.SDSQLOAD".      */
/*****/
/* Specify attribute list for dataset allocations          */
/*****/
ATTR PRINTDCB  LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB  LRECL(80)  RECFM(F B)   BLKSIZE(3120)
ATTR UDUMPDCB  LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB   LRECL(79)  RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO                                  */
/*****/
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets                                    */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQUCFRM) DA('QMF710.DSQUCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM')   SHR REUSE
ALLOC FI(ADMGDF)   DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS)  DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF                                  */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP)  SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQPILL)  NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT)  NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE)  DA('QMF710.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command            */
/*****/
CALL samFORT
EXIT CODE(0)

```

그림 55. ISPF 없이 TSO에서 프로그램을 실행하는 CLIST

FORTRAN 언어 인터페이스

ISPF하의 TSO에서 실행

다음 프로그램은 VS FORTRAN 컴파일러를 사용하여 호출 인터페이스 응용프로그램을 실행합니다. 일부 매개변수는 설치에 따라 달라질 수 있습니다. 자세한 내용은 QMF 관리자에게 문의하십시오.

```

PROC 0
CONTROL ASIS
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF710.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD', +
        'PRODUCT.VSF2LOAD')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF710.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF710.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF710.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQUCFRM) DA('QMF710.DSQUCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF710.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(samFORT) NEWAPPL(DSQE)
EXIT CODE(4)

```

그림 56. ISPF하의 TSO에서 프로그램을 실행하는 CLIST

EXIT CODE(4)는 ISPF 특성 패널이 표시되지 않도록 합니다.

PL/I 언어 인터페이스

PL/I 호출 인터페이스는 다른 SAA 언어에 대해 제공된 인터페이스와 일치합니다.

CICS에서 QMF와 함께 사용할 PL/I의 최소한의 릴리스 레벨은 PL/I 버전 2입니다. PL/I 버전 2는 VSE/ESA에서 지원되지 않습니다.

PL/I의 인터페이스 통신 영역 맵핑(DSQCOMML)

DSQCOMML는 DSQCOMM 맵핑을 PL/I에 제공하고 해당 제품과 함께 제공됩니다. 표19에서는 DSQCOMML의 값을 보여줍니다.

표 19. DSQCOMML의 인터페이스 통신 영역

구조 이름	데이터 유형	설명
DSQ_RETURN_CODE	FIXED BIN(31)	QMF 명령이 실행된 후의 상태를 나타냅니다. 해당 값은 다음과 같습니다. DSQ_SUCCESS 요청의 성공적인 실행 DSQ_WARNING 경고 메시지를 나타내는 정상적인 완료 DSQ_FAILURE 명령이 제대로 실행되지 않음 DSQ_SEVERE 심각한 오류. QMF 세션이 종료됨
DSQ_INSTANCE_ID	FIXED BIN(31)	START 명령의 실행 동안 QMF에 의해 설정된 ID
DSQ_COMM_LEVEL	CHAR(12)	DSQCOMM의 레벨을 식별합니다. 이것을 DSQ_CURRENT_COMM_LEVEL 값으로 설정한 다음 QMF START 명령을 발행합니다.
DSQ_PRODUCT	CHAR(2)	사용중인 IBM 조회 제품을 식별합니다.
DSQ_PRODUCT_RELEASE	CHAR(2)	사용중인 조회 제품의 릴리스 레벨을 식별합니다.
DSQ_RESERVE1	CHAR(28)	앞으로 사용되도록 예약됨
DSQ_MESSAGE_ID	CHAR(8)	완료 메시지 ID
DSQ_Q_MESSAGE_ID	CHAR(8)	조회 메시지 ID
DSQ_START_PARM_ERROR	CHAR(8)	START가 매개변수 오류로 인해 실패했을 경우 오류 상태에 있는 매개변수

표 19. DSQCOMML의 인터페이스 통신 영역 (계속)

구조 이름	데이터 유형	설명
DSQ_CANCEL_IND	CHAR(1)	QMF 명령이 실행되는 동안 사용자가 취소했을 경우 다음 두 값 중 하나를 포함합니다. DSQ_CANCEL_YES DSQ_CANCEL_NO
DSQ_RESERVE2	CHAR(23)	앞으로 사용되도록 예약됨
DSQ_RESERVE3	CHAR(156)	앞으로 사용되도록 예약됨
DSQ_MESSAGE_TEXT	CHAR(128)	완료 메시지 텍스트
DSQ_Q_MESSAGE_TEXT	CHAR(128)	조회 메시지 텍스트

PL/I의 함수 호출

QMF는 두 함수 호출(DSQCIPL 및 DSQCIPX)을 PL/I에 제공합니다. 두 호출 모두 통신 매크로 DSQCOMML에서 설명됩니다.

DSQCIPL 구문

이 호출은 응용프로그램 변수에 액세스를 요구하지 않는 QMF 명령에 대한 것입니다. 대부분의 QMF 명령에서 이 호출을 사용하십시오.

```
CALL DSQCIPL(DSQCOMM,
             CMDLTH,
             CMDSTR)
```

매개변수 값은 다음과 같습니다.

DSQCOMM

인터페이스 통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이

CMDSTR

실행할 QMF 명령, CMDLTH에 의해 지정된 길이의 대문자 문자열

DSQCIPX 구문

이 호출은 응용프로그램 변수에 액세스를 요구하는 세 명령(START, GET GLOBAL 및 SET GLOBAL의 확장 형식)에 대한 것입니다.

PL/I 언어 인터페이스

이 호출에 대한 구문은 다음과 같습니다.

```
CALL DSQCIPX(DSQCOMM,  
            CMDLTH,  
            CMDSTR,  
            PNUM,  
            KLTH,  
            KWORD,  
            VLTH,  
            VALUE,  
            VTYPE)
```

매개변수 값은 다음과 같습니다.

DSQCOMM

인터페이스 통신 영역

CMDLTH

명령 문자열 CMDSTR의 길이. 이 값은 정수 FIXED BIN(31) 매개변수입니다.

CMDSTR

실행할 QMF 명령. 이 값은 CMDLTH에 의해 지정된 길이의 대문자 문자열입니다.

PNUM

명령 키워드의 수. 이 값은 정수 FIXED BIN(31) 매개변수입니다.

KLTH

각 지정된 키워드의 길이. 이 값은 정수 FIXED BIN(31) 매개변수 또는 매개변수 배열입니다.

KWORD

QMF 키워드. 각각은 문자 또는 KLTH에 의해 지정된 것과 같은 길이를 가지는 문자 또는 문자의 구조입니다. 모든 키워드가 같은 길이이면 문자의 배열을 사용할 수 있습니다. QMF는 키워드가 연속적인 기억영역에 있고 어떤 특수 분리 문자에 의해 분리되지 않는다고 가정합니다.

VLTH

키워드와 연관된 각 값의 길이. 이 값은 정수 FIXED BIN(31) 매개변수 또는 매개변수 배열입니다.

VALUE

각 키워드와 연관된 값. 유형은 VTYPE 매개변수에서 지정되어 있으며 문자, 문자의 구조, 정수 FIXED BIN(31) 매개변수 또는 매개변수 배열이 될 수 있습니다. 문자값이 있으면 QMF는 값이 연속적인 기억영역에 있고 어떤 특수 분리 문자에 의해 분리되지 않는다고 가정합니다.

VTTYPE

값 문자열 VALUE의 QMF 데이터 유형. VTYPE는 두 값 중에 하나를 가질 수 있으며 이 값은 통신 매크로 DSQCOMML에서 제공됩니다.

문자 값의 경우 DSQ_VARIABLE_CHAR.

정수 FIXED BIN(31) 값의 경우 DSQ_VARIABLE_FINT

VALUE 필드에서 지정된 모든 값은 VTYPE에서 지정된 데이터 유형을 가집니다.

MVS에서 CICS 사용자를 위한 마이그레이션 정보

DSQCIPL 및 DSQCIPLX 호출은 버전 3 릴리스 1 수정판 1에서 버전 3 릴리스 2로 변경되었습니다. QMF 제공 함수 호출과 주 QMF 프로그램간의 인터페이스는 CALL 인터페이스에서 EXEC CICS LINK 인터페이스로 변경되었습니다. 새 인터페이스는 사용자 프로그램과 QMF 제품을 확실하게 구별해 줍니다. 인터페이스가 변경되었기 때문에 버전 3 릴리스 1 또는 이전의 프로그램으로부터 마이그레이션하려면 호출 인터페이스를 사용하는 프로그램을 다시 링크 편집해야 합니다.

PL/I 프로그래밍 예제

다음 샘플 프로그램 DSQABFP는 QMF와 함께 제공되고 IBM PL/I를 사용합니다.

여기에서 샘플 소스 코드 목록을 볼 수 있고 온라인으로 액세스할 수도 있습니다.

- VM의 경우, 샘플 프로그램은 제품 디스크에 있습니다.
- OS/390의 경우, 샘플 프로그램은 라이브러리 QMF710.SDSQSAPE의 멤버입니다.
- CICS에서 QMF와 함께 사용하는 경우, 요구되는 PL/I의 최소의 릴리스 레벨은 버전 2입니다. PL/I 버전 2는 VSE/ESA에서 제공되지 않습니다.

PL/I 언어 인터페이스

PL/I 호출 인터페이스용 샘플 프로그램은 다음 기능을 수행합니다.

- QMF를 시작합니다.
- 세 개의 글로벌 변수를 설정합니다.
- Q1이라는 조회를 실행합니다.
- 서식 F1을 사용하여 결과 보고서를 인쇄합니다.
- QMF 세션을 종료합니다.

QMF는 조회 Q1 또는 서식 F1을 제공하지 않으나, 샘플 프로그램은 이 오브젝트를 사용합니다.

또한 이 절에서는 호출 인터페이스를 사용하여 PL/I 프로그램을 컴파일하고, 링크 편집하고, 실행하는 방법도 보여줍니다. QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

```
DSQABFP: PROCEDURE OPTIONS(MAIN REENTRANT) REORDER;                00001000
/*****                                                              00002000
/* Sample Program: DSQABFP                                          */ 00003000
/* PL/I Version of the SAA Query Callable Interface                 */ 00004000
/*****                                                              00005000
                                                                00006000
/*****                                                              00007000
/* Include and declare query interface communications area         */ 00008000
/*****                                                              00009000
%INCLUDE SYSLIB(DSQCOMML);                                         00010000
                                                                00011000
/*****                                                              00012000
/* Builtin function                                               */ 00013000
/*****                                                              00014000
DCL LENGTH BUILTIN;                                               00015000
                                                                00016000
/*****                                                              00017000
/* Query interface command length and commands                    */ 00018000
/*****                                                              00019000
DCL COMMAND_LENGTH        FIXED BIN(31);                          00020000
DCL START_QUERY_INTERFACE CHAR(5)  INIT('START');                00021000
DCL SET_GLOBAL_VARIABLES CHAR(10)  INIT('SET GLOBAL');           00022000
DCL RUN_QUERY             CHAR(12)  INIT('RUN QUERY Q1');         00023000
DCL PRINT_REPORT         CHAR(22)  INIT('PRINT REPORT (FORM=F1)'); 00024000
DCL END_QUERY_INTERFACE  CHAR(4)   INIT('EXIT');                 00025000
                                                                00026000
/*****                                                              00027000
/* Query command extension, number of parameters and lengths     */ 00028000
/*****                                                              00029000
DCL NUMBER_OF_PARAMETERS  FIXED BIN(31);/* number of variables   */ 00030000
DCL KEYWORD_LENGTHS(10)   FIXED BIN(31);/* lengths of keyword names*/ 00031000
DCL DATA_LENGTHS(10)     FIXED BIN(31);/* lengths of variable data*/ 00032000
                                                                00033000
```

그림 57. 샘플 PL/I 프로그램(DSQABFP) (1/3)


```

/*****/ 00034000
/* Keyword parameter and value for START command */ 00035000
/*****/ 00036000
DCL START_KEYWORDS CHAR(8) INIT('DSQSMODE'); 00037000
DCL START_KEYWORD_VALUES CHAR(11) INIT('INTERACTIVE'); 00038000
00039000

/*****/ 00040000
/* Keyword parameter and value for SET command */ 00041000
/*****/ 00042000
DCL 1 SET_KEYWORDS, 00043000
    3 SET_KEYWORDS_1 CHAR(7) INIT('MYVAR01'), 00044000
    3 SET_KEYWORDS_2 CHAR(5) INIT('SHORT'), 00045000
    3 SET_KEYWORDS_3 CHAR(7) INIT('MYVAR03'); 00046000
00047000

DCL 1 SET_VALUES, 00048000
    3 SET_VALUES_1 FIXED BIN(31), 00049000
    3 SET_VALUES_2 FIXED BIN(31), 00050000
    3 SET_VALUES_3 FIXED BIN(31); 00051000
00052000

/*****/ 00053000
/* Main program */ 00054000
/*****/ 00055000
DSQCOMM = ''; 00056000
DSQ_COMM_LEVEL = DSQ_CURRENT_COMM_LEVEL; 00057000
00058000

/*****/ 00059000
/* Start a query interface session */ 00060000
/*****/ 00061000
NUMBER_OF_PARAMETERS = 1; 00062000
COMMAND_LENGTH = LENGTH(START_QUERY_INTERFACE); 00063000
KEYWORD_LENGTHS(1) = LENGTH(START_KEYWORDS); 00064000
DATA_LENGTHS(1) = LENGTH(START_KEYWORD_VALUES); 00065000
00066000

CALL DSQCIPX(DSQCOMM, 00067000
    COMMAND_LENGTH, 00068000
    START_QUERY_INTERFACE, 00069000
    NUMBER_OF_PARAMETERS, 00070000
    KEYWORD_LENGTHS, 00071000
    START_KEYWORDS, 00072000
    DATA_LENGTHS, 00073000
    START_KEYWORD_VALUES, 00074000
    DSQ_VARIABLE_CHAR); 00075000
00076000

```

그림 57. 샘플 PL/I 프로그램(DSQABFP) (2/3)

```

/*****/ 00077000
/* Set numeric values into query using SET command */ 00078000
/*****/ 00079000
NUMBER OF PARAMETERS = 3; 00080000
COMMAND_LENGTH = LENGTH(SET GLOBAL VARIABLES); 00081000
KEYWORD_LENGTHS(1) = LENGTH(SET_KEYWORDS_1); 00082000
KEYWORD_LENGTHS(2) = LENGTH(SET_KEYWORDS_2); 00083000
KEYWORD_LENGTHS(3) = LENGTH(SET_KEYWORDS_3); 00084000
DATA_LENGTHS(1) = 4; 00085000
DATA_LENGTHS(2) = 4; 00086000
DATA_LENGTHS(3) = 4; 00087000
SET_VALUES_1 = 20; 00088000
SET_VALUES_2 = 40; 00089000
SET_VALUES_3 = 84; 00090000
00091000
CALL DSQCIPX(DSQCOMM, 00092000
              COMMAND_LENGTH, 00093000
              SET_GLOBAL_VARIABLES, 00094000
              NUMBER_OF_PARAMETERS, 00095000
              KEYWORD_LENGTHS, 00096000
              SET_KEYWORDS, 00097000
              DATA_LENGTHS, 00098000
              SET_VALUES, 00099000
              DSQ_VARIABLE_FINT); 00100000
00101000
/*****/ 00102000
/* Run a Query */ 00103000
/*****/ 00104000
COMMAND_LENGTH = LENGTH(RUN_QUERY); 00105000
00106000
CALL DSQCIPL(DSQCOMM, 00107000
             COMMAND_LENGTH, 00108000
             RUN_QUERY); 00109000
00110000
/*****/ 00111000
/* Print the results of the query */ 00112000
/*****/ 00113000
COMMAND_LENGTH = LENGTH(PRINT_REPORT); 00114000
00115000
CALL DSQCIPL(DSQCOMM, 00116000
             COMMAND_LENGTH, 00117000
             PRINT_REPORT); 00118000
00119000
/*****/ 00120000
/* End the query interface session */ 00121000
/*****/ 00122000
COMMAND_LENGTH = LENGTH(END_QUERY_INTERFACE); 00123000
00124000
CALL DSQCIPL(DSQCOMM, 00125000
             COMMAND_LENGTH, 00126000
             END_QUERY_INTERFACE); 00127000
00128000
END DSQABFP; 00129000

```

그림 57. 샘플 PL/I 프로그램(DSQABFP) (3/3)

PL/I용 DSQCOMM

```

/*****/ 00001000
/* PL/I include for Query Callable Interface (MVS/VM) */ 00002000
/*****/ 00003000
00004000
/* Structure declare for Communications Area */ 00005000
DCL 00006000
1 DSQCOMM, 00007000
3 DSQ_RETURN_CODE FIXED BIN(31), /* function return code */ 00008000
3 DSQ_INSTANCE_ID FIXED BIN(31), /* start ID */ 00009000
3 DSQ_COMM_LEVEL CHAR(12), /* communications level */ 00010000
3 DSQ_PRODUCT CHAR(2), /* query product id */ 00011000
3 DSQ_PRODUCT_RELEASE CHAR(2), /* query product release */ 00012000
3 DSQ_RESERVE1 CHAR(28), /* reserved */ 00013000
3 DSQ_MESSAGE_ID CHAR(8), /* completion message ID */ 00014000
3 DSQ_Q_MESSAGE_ID CHAR(8), /* query message ID */ 00015000
3 DSQ_START_PARM_ERROR CHAR(8), /* start parms in error */ 00016000
3 DSQ_CANCEL_IND CHAR(1), /* cmd cancel indicator */ 00017000
/* 1 = cancelled, 0 = not cancelled*/ 00018000
3 DSQ_RESERVE2 CHAR(23), /* reserved */ 00019000
3 DSQ_RESERVE3 CHAR(156), /* reserved */ 00020000
3 DSQ_MESSAGE_TEXT CHAR(128), /* QMF command message */ 00021000
3 DSQ_Q_MESSAGE_TEXT CHAR(128); /* QMF query message */ 00022000
00023000
/* Return Codes */ 00024000
DCL 00025000
DSQ_SUCCESS FIXED BIN(31) INIT(0) STATIC, 00026000
DSQ_WARNING FIXED BIN(31) INIT(4) STATIC, 00027000
DSQ_FAILURE FIXED BIN(31) INIT(8) STATIC, 00028000
DSQ_SEVERE FIXED BIN(31) INIT(16) STATIC; 00029000
00030000
/* Communications Level */ 00031000
DCL 00032000
DSQ_CURRENT_COMM_LEVEL CHAR(12) INIT('DSQL>001002<') STATIC; 00033000
00034000
/* Query Product ID */ 00035000
DCL 00036000
DSQ_QRW CHAR(2) INIT('01') STATIC, 00037000
DSQ_QMF CHAR(2) INIT('02') STATIC, 00038000
DSQ_QM4 CHAR(2) INIT('03') STATIC; 00039000
00040000

```

그림 58. PL/I 통신 영역(DSQCOMML) (1/2)

PL/I 언어 인터페이스

```

/* Query Product Release ID                                */ 00041000
DCL                                                         00042000
  DSQ_QRW_V1R2      CHAR(2) INIT('01') STATIC,           00043000
  DSQ_QRW_V1R3      CHAR(2) INIT('02') STATIC,           00044000
  DSQ_QMF_V2R4      CHAR(2) INIT('01') STATIC,           00045000
  DSQ_QMF_V3R1      CHAR(2) INIT('02') STATIC,           00046000
  DSQ_QMF_V3R1M1    CHAR(2) INIT('03') STATIC,           00047000
  DSQ_QMF_V3R2      CHAR(2) INIT('04') STATIC,           00048000
  DSQ_QMF_V3R3      CHAR(2) INIT('05') STATIC,           00049000
  DSQ_QMF_V6R1      CHAR(2) INIT('06') STATIC,           00050000
  DSQ_QM4_V1R1      CHAR(2) INIT('01') STATIC;           00051000
                                                         00052000

/* Cancelled Indicator                                    */ 00053000
DCL                                                         00054000
  DSQ_CANCEL_YES    CHAR(1) INIT('1') STATIC,             00055000
  DSQ_CANCEL_NO     CHAR(1) INIT('0') STATIC;             00056000
                                                         00057000

/* Variable Types                                         */ 00058000
DCL                                                         00059000
  DSQ_VARIABLE_CHAR CHAR(4) INIT('CHAR') STATIC,          00060000
  DSQ_VARIABLE_FINT CHAR(4) INIT('FINT') STATIC;          00061000
                                                         00062000

/* Mode                                                    */ 00063000
DCL                                                         00064000
  DSQ_INTERACTIVE   CHAR(1) INIT('1') STATIC,             00065000
  DSQ_BATCH          CHAR(1) INIT('2') STATIC;             00066000
                                                         00067000

/* Yes or No                                              */ 00068000
DCL                                                         00069000
  DSQ_YES            CHAR(1) INIT('1') STATIC,             00070000
  DSQ_NO             CHAR(1) INIT('2') STATIC;             00071000
                                                         00072000

/* Query Interface Entry Point                            */ 00073000
DCL                                                         00074000
  DSQCIPL ENTRY (*, /* interface block                    */ 00075000
                  FIXED BIN(31), /* length of command */ 00076000
                  CHAR(*)) /* command string            */ 00077000
                  EXTERNAL OPTIONS(ASSEMBLER);           00078000
                                                         00079000

  DSQCIPX ENTRY (*, /* interface block                    */ 00080000
                 FIXED BIN(31), /* length of command */ 00081000
                 CHAR(*), /* command string            */ 00082000
                 FIXED BIN(31), /* # of command keywords */ 00083000
                 *, /* length of keyword          */ 00084000
                 *, /* keyword string              */ 00085000
                 *, /* length of value              */ 00086000
                 *, /* value of keyword            */ 00087000
                 CHAR(4)) /* data type of value         */ 00088000
                 EXTERNAL OPTIONS(ASSEMBLER);           00089000

```

그림 58. PL/I 통신 영역(DSQCOMML) (2/2)

CICS에서 프로그램 실행

프로그램을 작성한 후에는 컴파일하고 실행해야 합니다. 이 절에 나열된 예제에서 는 반드시 수행해야 할 단계를 보여줍니다.

QMF는 이 예제에서 REXX EXEC, JCL 또는 CLIST를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

MVS의 CICS에서 변환, 컴파일 및 링크 편집

QMF 호출 인터페이스를 사용하는 프로그램을 변환하고, 컴파일하고, 링크 편집할 경우 다음을 고려하십시오.

- 통신 영역 DSQCOMML은 컴파일 단계에서 사용가능해야 하며 혹은 프로그램에 복사해야 합니다.
- QMF 인터페이스 모듈 DSQCIPL 및 DSQCIPX를 프로그램의 링크 편집 단계 동안 사용할 수 있어야 합니다.

다음은 CICS 제공 프로시저 DFHEBTPL을 사용하는 예제입니다. 이 프로시저 사용 방법에 관한 지침은 사용하는 릴리스에 대한 *CICS for VSE/ESA System Definition Guide*를 참조하십시오.

```
//samPLI JOB
// EXEC PROC=DFHEBTPL
//TRN.SYSIN DD *
*PROCESS XOPTS(CICS translator options .....
      .
      Your program or copy of QMF sample DSQABFP
      .
/*
/** Provide Access to QMF Communications Macro DSQCOMML
//PLI.SYSLIB DD DSN=QMF710.SDSQSAPE,DISP=SHR
/** Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF710.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
      INCLUDE CICSLOAD(DFHPL10I)
      INCLUDE CICSLOAD(DFHEPI)
      INCLUDE QMFLOAD(DSQCIPX)
      INCLUDE QMFLOAD(DSQCIPX)
      ORDER DFHPL10I,DFHEPI
      ENTRY sampPLI
      MODE AMODE(31) RMODE(ANY)
      NAME sampPLI(R)
/*
```

그림 59. CICS 변환기, PL/I 컴파일러 및 연계 편집기를 실행하는 JCL

VSE의 CICS에서 변환, 컴파일 및 링크 편집

그림60에서 VSE 작업 제어는 PL/I 프로그램을 VSE에서 실행되는 CICS로 설치하는 예제입니다. QMF와 함께 제공되는 이 예제는 DSQ3CIP.Z라는 이름으로 QMF 서브라이브러리에 들어 있습니다. 자세히 알려면 *CICS for VSE/ESA System Definition Guide* 및 *PL/I VSE Programming Guide* 내용을 참조하십시오.

```

..* $$ JOB JNM=DSQ3CIP,DISP=D,CLASS=0
// JOB DSQ3CIP      Sample job to Install QMF Callable Interface (PL/I)
* -----
* Install QMF Callable Interface Example (PL/I)
* -----
// SETPARM VOLID=valid      *-- update valid for syspchg
// SETPARM START=rtrk      *-- update start track/block (syspchg)
// SETPARM SIZE=ntrks      *-- update number of tracks/blocks (syspchg)
* -----
// DLBL   IJSYSPH,'CICS.TRANSLAT.OUTPUT',0
// EXTENT SYSPCH,,1,0,&START,&SIZE
ASSGN SYSPCH,DISK,VOL=&VOLID,SHR
* Library search chain must contain the QMF, CICS and PL/I sublibrary
// LIBDEF *,SEARCH=(PRD2.PROD,PRD1.BASE,PRD2.CONFIG)
// LIBDEF PHASE,CATALOG=PRD2.PROD
* -----
* Step 1: Translate callable interface program (PL/I)
* -----
* You may need to update or remove the SLI statement for your program.
* -----
// EXEC   DFHEPPI$,SIZE=256K,PARM='XOPTS(CICS)'
..* $$ SLI MEM=DSQABFP.Z,S=PRD2.PROD
/*
* -----
* Step 2: Compile callable interface program (PL/I)
* -----
CLOSE SYSPCH,00D
// DLBL   IJSYSIN,'CICS.TRANSLAT.OUTPUT',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLID,SHR
// OPTION NODECK,CATAL
// PHASE DSQABFP,*
// INCLUDE DFHPL1I
// EXEC PLIOPT
CLOSE SYSIPT,SYSRDR
/*

```

그림 60. VSE의 샘플 JCL (1/2)

```

* -----
* Step 3: Link-edit callable interface program (PL/I)
* -----
// EXEC LNKEDT,PARM='AMODE=31,RMODE=ANY'
/*
/ &
// JOB RESET
ASSGN SYSIPT,SYSRDR      IF 1A93D, CLOSE SYSIPT,SYSRDR
ASSGN SYSPCH,00D        IF 1A93D, CLOSE SYSPCH,00D
/ &
..* $$ E0J

```

그림 60. VSE의 샘플 JCL (2/2)

VM의 CMS에서 프로그램 컴파일 및 실행

다음 프로그램은 PL/I 컴파일러를 사용하여 호출 인터페이스 응용프로그램을 컴파일하고 실행합니다.

QMF는 이 예제에서 REXX EXEC를 제공하지 않으나, 여기에 복사하여 설치에 맞게 고칠 수 있습니다.

PL/I 언어 인터페이스

```

/*****/
/* Compile QMF PL/I program and run it. */
/*****/
TRACE off
ADDRESS CMS

/*****/
/* Access PL/I product disk using a program, PRODUCT, that you */
/* write. */
/*****/
"EXEC PRODUCT PLIV"

/*****/
/* Get QMF DSQCOMM into a macro library and set GLOBAL compile */
/* maclibs. */
/*****/
"ERASE TEMPP MACLIB A"
"MACLIB GEN TEMPP DSQCOMML"
Maclist = "TEMPP PLICOMP DMSSP CMSLIB OSMACRO"
"GLOBAL MACLIB" Maclist

/*****/
/* Compile the program */
/*****/
POPTS = '(INC SOURCE LIST LMSG M NAG NC(E) NIS NOESD NSTG OPT(2)'
'PLIOPT' yourname popts

/*****/
/* Access SQL/DS and initialize database */
/*****/
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"

/*****/
/* Access GDDM product disk */
/*****/
"EXEC PRODUCT GDDM"
```

그림 61. 프로그램을 컴파일하고 실행하는 REXX 프로그램 (1/2)


```

/*****/
/* Issue Filedefs for QMF product */
/*****/
"FILEDEF ISPLLIB CLEAR"
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A ( PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 ( PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * ( PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQDLIB LOADLIB *"

/*****/
/* Provide access to QMF and PL/I program libraries */
/*****/
'GLOBAL MACLIB TEMPP'
'GLOBAL LOADLIB DSQDLIB PLILIB'
'GLOBAL TXTLIB PLILIB IBMLIB ADMRLIB ADMPLIB ADMGLIB'

```

Say "Starting to run PL/I program"
"RUN yourname"

Exit 0

그림 61. 프로그램을 컴파일하고 실행하는 REXX 프로그램 (2/2)

설치에 맞게 이 프로그램을 변경해야 합니다.

TSO에서 컴파일 및 링크 편집

다음 작업은 PL/I 컴파일러를 사용하여 호출 인터페이스 응용프로그램을 컴파일하고 응용프로그램을 링크 편집합니다. 일부 매개변수는 설치에 따라 달라질 수 있습니다. 자세한 내용은 QMF 관리자에게 문의하십시오.

PL/I 언어 인터페이스

```
//samPLI    JOB
//STEP1     EXEC IEL1CL
//* Provide Access to QMF Communications Macro DSQCOMML
//PLI.SYSLIB DD DSN=QMF710.SAMPLIB,DISP=SHR
//PLI.SYSIN  DD *
            .
            Your program or copy of QMF sample DSQABFP
            .
/*
/** Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF710.SDSQLOAD,DISP=SHR
//LKED.SYSIN  DD *
            INCLUDE QMFLOAD(DSQCIPL)
            INCLUDE QMFLOAD(DSQCIPX)
            ENTRY sampPLI
            MODE AMODE(31) RMODE(ANY)
            NAME  sampPLI(R)
/*
```

그림 62. PL/I 컴파일러 및 연계 편집기를 실행하는 JCL

ISPF 없이 TSO에서 실행

TSO 환경에 맞게 프로그램을 컴파일한 후에 다음 CLIST가 프로그램을 실행합니다.

```

PROC 0
CONTROL ASIS
/*****/
/* Note: QMF, DB2, GDDM and PL/I load libraries must be */
/* allocated before running this CLIST. */
/* Name of QMF load library is "QMF710.SDSQLOAD". */
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF710.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF710.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command */
/*****/
CALL sampPLI
EXIT CODE(0)

```

그림 63. ISPF 없이 TSO에서 프로그램을 실행하는 CLIST

ISPF하의 TSO에서 실행

TSO 환경에 맞게 프로그램을 컴파일한 후에 다음 CLIST가 프로그램을 실행합니다.

REXX 언어 인터페이스

```
PROC 0
CONTROL ASIS
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF710.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF710.SDSQEXCE')
/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF710.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD', +
        'PLI.PLILINK','PLI.SIBMLINK')
ALLOC FI(ISPLIB) SHR REUSE +
    DA('QMF710.SDSQMLBE','ISR.ISRMLIB','ISP.ISPLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF710.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF710.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF710.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF710.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRML) DA('QMF710.DSQCFRML') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF710.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(sampPLI) NEWAPPL(DSQE)
EXIT CODE(4)
```

그림 64. ISPF하의 TSO에서 프로그램을 실행하는 CLIST

EXIT CODE(4)는 ISPF 특성 패널이 표시되지 않도록 합니다.

REXX 언어 인터페이스

여기에서 제공된 REXX 호출 인터페이스는 다른 SAA 언어에 대해 제공된 인터페이스와 일치합니다.

CICS 사용자를 위한 주의사항

REXX는 QMF CICS하에서 사용할 수 없으므로 REXX용 QMF 호출 인터페이스도 CICS에서 작동되지 않습니다.

REXX는 해석 언어이므로 컴파일할 필요가 없습니다. 그러나 컴파일된 REXX 또는 컴파일된 다른 언어를 사용하여 작성된 프로그램은 해석 REXX를 사용하여 작성된 같은 프로그램보다 성능이 더 좋습니다. REXX 컴파일러는 REXX 프로그램용으로 사용가능하지만, 논리를 갖는 프로시저의 경우에는 사용할 수 없습니다.

TSO에서 REXX 호출 인터페이스를 사용하여 QMF를 호출함으로써 논리를 갖는 프로시저와 특정 서식 기능(계산, 정의된 열, 조건)을 사용할 때 REXX 서비스를 사용하는 데 필요한 자원을 줄일 수 있습니다. 그 이유는 이 모든 기능이 REXX를 사용하기 때문입니다.

예를 들어, QMF 세션이 REXX 호출 인터페이스를 사용하여 초기화되면 REPORT에 있는 PRINT REPORT 또는 BOTTOM을 실행하는 데 더 작은 자원이 필요합니다. TSO/E에서 QMF를 실행하는 데 있어 자원 소비가 줄어든다는 것은 매우 중요하고 주목할만한 일입니다.

REXX 언어는 명령이 처리되는 위치와 방법을 결정하는 명령 환경에서 작동합니다. QMF 명령을 발행하는 REXX 프로그램을 작성하면 ADDRESS QRW 명령을 통해 QMF 명령 환경을 사용할 수 있습니다. 자세한 내용은 281 페이지의 『부록D. ADDRESS QRW: QMF 명령 환경 사용』을 참조하십시오.

REXX의 인터페이스 통신 변수

통신 변수는 다음 REXX 변수로 구성되어 있습니다. 이러한 변수들은 모든 호출이 완료된 후에 설정됩니다.

REXX 언어 인터페이스

표20에서는 호출 프로그램에 의해 변경되면 안 되는 인터페이스 통신 변수를 보여줍니다.

표 20. RWXX의 인터페이스 통신 변수

구조 이름	설명
dsq_return_code	SAA 조회의 상태를 표시하는 정수. 가능한 값은 다음과 같습니다. dsq_success 요청의 성공적인 처리 dsq_warning 경고 메시지를 나타내는 정상적인 완료 dsq_failure 명령이 올바르게 처리되지 않음 dsq_severe 심각한 오류. SAA 조회 세션이 종료됨. SAA 조회 세션이 종료되었기 때문에 이 인스턴스 ID를 사용하여 SAA 조회의 추가 호출을 수행할 수 없습니다. dsq_return_code의 값도 REXX 변수 rc에 배치됩니다.
dsq_instance_id	START 명령의 처리 동안 SAA 조회에 의해 설정되는 ID
dsq_product	사용중인 조회 관리자 제품. 가능한 값은 다음과 같습니다. dsq_qrw OS/2 조회 관리자 dsq_qmf QMF dsq_qm4 OS/400 조회 관리

표 20. RWXX의 인터페이스 통신 변수 (계속)

구조 이름	설명
dsq_product_release	<p>사용중인 조회 제품의 릴리스 레벨. 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> OS/2: <ul style="list-style-type: none"> dsq_qrw_v1r2 버전 1 릴리스 2 dsq_qrw_v1r3 버전 1 릴리스 3 OS/400: <ul style="list-style-type: none"> dsq_qm4_v1r4 버전 1 릴리스 1 QMF: <ul style="list-style-type: none"> dsq_qmf_v2r4 QMF 버전 2 릴리스 4 dsq_qmf_v3r1 QMF 버전 3 릴리스 1 dsq_qmf_v3r1m1 QMF 버전 3 릴리스 1 수정판 1 dsq_qmf_v3r2 QMF 버전 3 릴리스 2 dsq_qmf_v3r3 QMF 버전 3 릴리스 3 dsq_qmf_v6r1 QMF 버전 6 dsq_qmf_v7r1 QMF 버전 7
dsq_message_id	완료 메시지 ID
dsq_q_message_id	조회 메시지 ID
dsq_start_parm_error	START가 매개변수 오류로 인해 실패했을 경우 오류 상태에 있는 매개변수

REXX 언어 인터페이스

표 20. RWXX의 인터페이스 통신 변수 (계속)

구조 이름	설명
dsq_cancel_ind	명령 취소 표시기. QMF가 명령을 실행하고 있는 동안 사용자가 명령 처리를 취소했는지를 표시합니다. 가능한 값은 다음과 같습니다. dsq_cancel_yes 사용자가 명령을 취소했습니다. dsq_cancel_no 사용자가 명령을 취소하지 않았습니다.
dsq_message_text	완료 메시지 텍스트
dsq_q_message_text	조회 메시지 텍스트

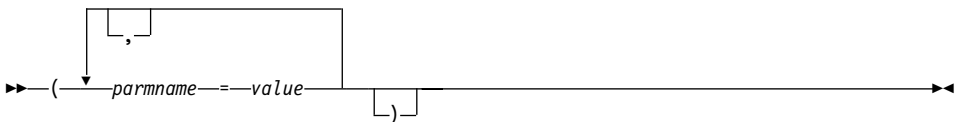
REXX에 대한 함수 호출

보통의 REXX 함수 호출을 사용하여 호출 인터페이스로 액세스할 수 있습니다. QMF는 DSQCIX라는 외부 서브루틴을 제공하며 이는 모든 SAA 조회 명령을 실행하는 데 사용됩니다.

DSQCIX 선형 구문

```
call DSQCIX cmd parmlist
```

- *cmd*는 대문자 문자열로 작성된 QMF 명령입니다.
- *parmlist*는 다음 차트에서 표시된 것처럼 매개변수와 값 쌍의 목록입니다.



*parmlist*를 포함하는 전체 명령은 문자열로 작성된 단일 RWXX 변수로서 QMF에 전달되어야 합니다. 이 문자열은 인용 부호 (') 또는 (" ")로 묶어야 합니다. 명령 문자열의 부분으로서 REXX 변수를 사용할 경우에는 인수를 인용 부호로 묶지 마십시오. 예를 들면, 다음과 같습니다.

```
CALL DSQCIX "RUN QUERY NAME (&ECN="REXAUG",CONFIRM=YES)"
```

parmname

매개변수의 이름

value

매개변수 이름과 연관되는 값은 parmname에 의해 지정됩니다.

예제:

```
call DSQCIX "RUN QUERY Q1"
call DSQCIX "PRINT REPORT (FORM=F1"
call DSQCIX "EXIT"
```

*parmlist*에서 다음 요소의 존재 여부와 관계없이 같은 결과가 발생합니다.

매개변수 사이의 쉼표(,)

하나의 공백은 같은 결과를 생성합니다.

오른쪽 괄호())

필요하지 않음

***parmname*과 *value* 사이의 등호(=)**

하나의 공백은 같은 결과를 생성합니다.

다음은 모두 같은 결과를 생성합니다.

```
call dsqcix "SET GLOBAL (abc=17, def=26"
call dsqcix "SET GLOBAL ( abc=17 def=26"
call dsqcix "SET GLOBAL ( abc=17 , def=26)"
call dsqcix "SET GLOBAL (abc 17 def=26)"
```

REXX 프로그래밍 예제

다음 프로그램 DSQABFX는 QMF와 함께 제공됩니다.

여기에 나열되는 샘플 소스 코드를 보거나 온라인으로 액세스할 수 있습니다. MVS의 경우 샘플 프로그램은 라이브러리 QMF710.SDSQEXCE의 멤버이며, VM의 경우 샘플 프로그램은 제품 디스크에 있습니다. REXX를 QMF CICS에서는 사용할 수 없습니다.

REXX 호출 인터페이스에 대한 샘플 프로그램은 다음 기능을 수행합니다.

- QMF를 시작합니다.
- 세 개의 글로벌 변수를 설정합니다.
- Q1이라는 조회를 실행합니다.
- 서식 F1을 사용하여 결과 보고서를 인쇄합니다.

REXX 언어 인터페이스

- QMF 세션을 종료합니다.

QMF는 조회 Q1 또는 서식 F1을 제공하지 않으나, 샘플 프로그램은 이 오버젝트를 사용합니다.

```
/*REXX*****  
/* Sample Program: DSQABFX *  
/* REXX Version of the SAA Query Callable Interface *  
/******  
  
/*****  
/* Start a query interface session *  
/*****  
  
call dsqcix "START (DSQSMODE=INTERACTIVE"  
say dsq_message_id dsq_message_text  
if dsq_return_code = dsq_severe then exit dsq_return_code  
  
/*****  
/* Set numeric values into query using SET command *  
/*****  
  
call dsqcix "SET GLOBAL (MYVAR01=20,SHORT=40,MYVAR03=84"  
say dsq_message_id dsq_message_text  
if dsq_return_code = dsq_severe then exit dsq_return_code  
  
/*****  
/* Run a Query *  
/*****  
  
call dsqcix "RUN QUERY Q1"  
say dsq_message_id dsq_message_text  
if dsq_return_code = dsq_severe then exit dsq_return_code  
  
/*****  
/* Print the results of the query *  
/*****  
  
call dsqcix "PRINT REPORT (FORM=F1)"  
say dsq_message_id dsq_message_text  
if dsq_return_code = dsq_severe then exit dsq_return_code  
  
/*****  
/* End the query interface session *  
/*****  
  
call dsqcix "EXIT"  
say dsq_message_id dsq_message_text  
exit dsq_return_code
```

그림 65. 샘플 REXX 프로그램(DSQABFX)

VM의 CMS에서 프로그램 실행

다음 프로그램은 REXX CALL 인터페이스를 사용하여 호출 인터페이스 응용프로그램을 실행합니다.

설치에 맞게 이 프로그램을 변경해야 합니다.

```

/***** */
/* Access SQL/DS and initialize database */
/***** */
"EXEC PRODUCT SQLDS" */
"EXEC SQLINIT DBNAME(SQLDBA)" */
*/
/***** */
/* Access GDDM product disk */
/***** */
"EXEC PRODUCT GDDM" */
*/
/***** */
/* Issue Filedefs for QMF product */
/***** */
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM" */
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 121 BLKSIZE 121 RECFM FBA PERM" */
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A ( PERM" */
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 ( PERM" */
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * ( PERM" */
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQDLIB LOADLIB *"

/***** */
/* Provide access to QMF and GDDM program libraries */
/***** */
"GLOBAL LOADLIB DSQDLIB "
"GLOBAL TXTLIB ADMRLIB ADMPLIB ADMGLIB"

/* The beginning of your REXX program ..... */
.
.
.
.
/* The end of your REXX program ..... */

```

그림 66. CMS에서 프로그램을 실행하는 REXX 프로그램

MVS의 TSO에서 프로그램 실행

다음과 유사한 프로그램 작성을 작성하여 REXX 프로그램을 실행할 수 있습니다.

```

/*****
/* Issue TSO Allocates for QMF Product */
/*****
Address TSO

"ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)"
"ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)"
"ATTR UDUMPCB LRECL(125) RECFM(V B A) BLKSIZE(1632)"
"ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)"
"ALLOC FI(SYSPROC) SHR REUSE ",
  "DA('QMF710.DSQCLSTE,',"
  "'DSN.DSNCLIST')"
```

```

"ALLOC FI(SYSEXEC) SHR REUSE ",
  "DA('QMF710.SDSQEXCE')"
```

```

"ALLOC FI(ISPLLIB) SHR REUSE ",
  "DA('QMF710.SDSQLOAD,',"
  "'ADM.GDDM.GDDMLOAD,',"
  "'DSN.DSNLOAD')"
```

```

"ALLOC FI(DSQPNLE) DA('QMF710.DSQPNLE') SHR REUSE"
"ALLOC FI(DSQPRINT) SYSOUT USING(PRINTDCB)"
"ALLOC FI(SYSPRT) SYSOUT(X) LRECL(132) RECFM(FBA) BLKSIZE(132)"
"ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)"
"ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPCB)"
"ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS"
"ALLOC DDNAME(DSQEDIT) UNIT(SYSDA) NEW USING(EDITDCB)"
"ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE"
"ALLOC FI(ADMGGMAP) DA('QMF710.DSQMAPE') SHR REUSE"
"ALLOC FI(ADMCFORM) DA('QMF710.DSQCHART') SHR REUSE"
"ALLOC FI(DSQUCFRM) DA('QMF710.DSQUCFRM') SHR REUSE"
"ALLOC FI(ADMGDF) DA('GDDM.ADMGDF') SHR REUSE"
"ALLOC FI(ADMSYMBL) DA('ADM.GDDM.GDDMSYM') SHR REUSE"

/* The beginning of your REXX program ..... */
.
.
.
/* The end of your REXX program ..... */
```

그림 67. TSO에서 프로그램을 실행하는 REXX 프로그램

INTERACT 루프 사용에 대한 REXX 예제

일반적으로 호출 인터페이스 프로그램이 INTERACT 명령을 발행하고 사용자가 END 명령을 발행하면 사용하는 프로그램의 작업 상태로 즉시 리턴됩니다. 그러나 대화식 QMF는 사용자가 END 명령을 발행하여 QMF 시작 패널로 리턴하도록 합니다. END 명령을 두 번 발행하면 QMF 세션은 종료됩니다.

다음 논리를 프로그램에 추가함으로써, 대화식 세션에서 END 명령을 작성할 수 있으며 이 세션은 대화식 QMF에서 END가 작동하는 방법과 유사하게 작동하는 호출 인터페이스 프로그램으로부터 INTERACT 명령에 의해 시작됩니다.

이 프로그램은 dsq_message_id를 사용하여 처리 방법을 판별합니다. 릴리스별로 이 값이 달라질 수 있습니다.

이 프로그램은 QMF로 분산되지 않습니다.

```

/*REXX*****
/* Sample Program: Using INTERACT loop */
*****
/* Start an interactive QMF session */
*****
trace error

parms = "START (DSQSMODE=INTERACTIVE"
call dsqcix parms
if dsq_return_code = dsq_severe then exit dsq_return_code
*****
/* SET GLOBAL to show panel IDs */
*****
call dsqcix "SET GLOBAL (DSQDC_SHOW_PANID=1"
if dsq_return_code = dsq_severe then exit dsq_return_code
*****
/* Issue message */
*****
call dsqcix "MESSAGE (TEXT='Ok, You may enter a command.')"
if dsq_return_code = dsq_severe then exit dsq_return_code
*****
/* INTERACT loop */
*****
Continue = "yes"
Do while continue = "yes"
  call DSQCIX "INTERACT"
  Select
    When (dsq_return_code = dsq_severe) Then /* Severe Error */
      Continue = "no"
    When (dsq_message_id = "DSQ21869") Then /* END from HOME panel */
      Continue = "no"
    When (dsq_message_id = "DSQ90557") Then /* User issued EXIT */
      Continue = "no"
    Otherwise nop /* OK continue session */
  End
End
*****
/* End the session */
*****
if dsq_message_id <> "DSQ90557" then /* EXIT not issued */
  call dsqcix "EXIT" /* Issue EXIT */
exit dsq_return_code

```

그림 68. INTERACT 루프를 사용하는 REXX 프로그램

부록B. 내보내기/가져오기 형식

이 장에서는 데이터의 QMF 형식에 대해 설명하고 암호화된 각 형식 오브젝트에 대한 표와 필드 수를 나열합니다.

- 서식
- 프롬프트 조회
- 보고서

이 목록의 설명과 예제에 대해서는 89 페이지의 『제8장 QMF 오브젝트 가져오기 및 내보내기』를 참조하십시오.

데이터에 대한 QMF 형식

EXPORT 명령(DATAFORMAT=QMF)을 사용하여 내보내는 데이터 파일은 레코드에서 데이터를 설명하는 헤더 레코드와 데이터를 포함하는 데이터 레코드의 두 부분으로 구성되어 있습니다.

헤더 레코드

외부 데이터 파일의 레코드 길이는 253 페이지의 『데이터 레코드』에서 설명된 것처럼 데이터 행의 길이입니다. 데이터 레코드 앞에 나오는 헤더 레코드 역시 이 길이로 분할됩니다. 표21은 헤더 레코드에 들어 있는 정보입니다.

표 21. 헤더 레코드 정보

바이트 위치	정보 및 유형
1-8	QMF 오브젝트 형식 레벨(8자의 데이터) QMF 오브젝트 형식 레벨은 특정 오브젝트에 대해 형식이 갱신되는 횟수를 지정합니다. QMF 1.0 이후 서식 오브젝트는 세 번 변경되었기 때문에 QMF 3.2에서 내보낸 서식의 오브젝트 형식 레벨은 4가 됩니다. 데이터 오브젝트 형식은 변경되지 않았기 때문에 여전히 오브젝트 형식 레벨 1에 있습니다.
9-10	헤더 레코드의 수(부호 있는 halfword 정수)
11-12	데이터 열의 수(부호 있는 halfword 정수)

내보내기/가져오기 형식

표 21. 헤더 레코드 정보 (계속)

바이트 위치	정보 및 유형
13-30, 37-54, ...	열 이름(18자의 데이터)
31-32, 55-56, ...	데이터 유형(부호 있는 halfword 정수). 데이터 유형 코드는 표22와 같이 표시됩니다.
33-34, 57-58, ...	<p>열 너비(부호 있는 halfword 정수). 대부분의 데이터 유형의 경우 바이트 단위의 열 너비입니다. 예외는 다음과 같습니다.</p> <ul style="list-style-type: none"> DECIMAL 열에서 중간 단어의 첫번째 바이트는 정밀도를 나타내고 두 번째 바이트는 배율을 나타냅니다. GRAPHIC 및 VARGRAPHIC 열에서 이 값은 2바이트 문자의 너비를 반영합니다. FLOAT 열에서 이 값은 단정도 부동 소수점 수를 나타내는 4이거나 배정도 부동 소수점 수를 나타내는 8 중 하나입니다.
35, 59, ...	널(null) 허용 여부: 널(null)이 허용되면 Y이고, 널(null)이 허용되지 않으면 N(1자의 데이터)
36, 60, ...	사용하지 않는 바이트

바이트 11-12는 열의 수를 나타냅니다. 이것은 13에서 36까지의 바이트 내의 정보가 헤더 레코드의 각 열에 대해 반복되는 것을 의미합니다. 헤더 레코드의 각 열은 24바이트를 요구합니다.

데이터 유형 코드는 표22와 같습니다.

표 22. 데이터 유형 코드

16진 코드	십진 코드	데이터 유형	의미
X'180'	384	DATE	날짜
X'184'	388	TIME	시간
X'188'	392	TIMESTAMP	시간소인
X'1C0'	448	VARCHAR	가변 문자
X'1C4'	452	CHAR	고정 문자
X'1D0'	464	VARGRAPHIC	가변 그래픽
X'1D4'	468	GRAPHIC	고정 그래픽
X'1E0'	480	FLOAT	부동 소수점
X'1E4'	484	DECIMAL	십진수
X'1F0'	496	INTEGER	정수
X'1F4'	500	SMALLINT	작은 정수

날짜, 시간, 시간소인 데이터를 항상 ISO 형식으로 내보냅니다.

데이터 유형의 형식에 대한 자세한 내용은 *DB2 UDB for OS390 SQL Reference* 를 참조하십시오.

데이터 레코드

데이터 레코드는 고정 길이 형식으로 되어 있고 내보낼 데이터를 포함합니다. 데이터 레코드에 가능한 최대 길이는 700바이트입니다. 데이터 레코드의 길이는 레코드를 구성하는 데이터 유형의 너비의 합계입니다. 각 데이터 유형의 너비를 계산하려면 다음 표를 사용하십시오.

중요: 최대 허용 길이가 255 이상인 VARCHAR 열이 있는 표는 내보낼 수 없습니다.

표 23. 코드화 형식 데이터 레코드에서 데이터 너비. 각 열에 바이트의 수를 추가함으로써 특정 데이터 유형의 너비를 계산합니다.

데이터 유형	널(null)		SO/SI	데이터
	표시기	길이 필드		
문자	2			헤더에서의 길이(LIH)
날짜	2			LIH
부동 소수점	2			8
정수	2			LIH
작은 정수	2			LIH
시간	2			LIH
시간소인	2			LIH
십진수	2			(정밀도 + 2) // 2
그래픽	2		2	(LIH × 2)
가변 문자	2	2		LIH
가변 그래픽	2	2	2	(2 × LIH)

주: LIH는 해당 열에 대한 헤더 레코드에 주어지는 너비입니다.

모든 데이터 레코드에는 2바이트의 표시기 정보가 있으며, 다음 값과 의미를 가질 수 있습니다.

값	의미
X'0000'	열은 유효한 데이터를 포함합니다.

내보내기/가져오기 형식

값	의미
X'FFFF'	열은 널 값을 포함합니다. 열에 있는 데이터는 의미가 없습니다.
X'FFFE'	

프롬프트 조회 오브젝트의 표 및 필드 번호

다음 표에서는 프롬프트 조회 내보내기 형식으로 각 표를 설명하는 T 레코드에 대한 프롬프트 조회 표 및 필드 번호를 보여줍니다. DESCRIPTION 열의 정보는 프롬프트 조회 기준 패널에서 특정 필드를 식별합니다.

표 정의(필드 번호 1110)를 항상 내보냅니다. 둘 이상의 표가 선택되면 결합 상태(필드 번호 1510)를 항상 내보냅니다.

프롬프트 조회 파일을 가져오려면, 파일에는 표 T 레코드 앞에 H 레코드가 있어야 합니다. 표를 지정할 필요는 없습니다. 표가 지정되지 않으면 빈 조회가 가져오기됩니다. 둘 이상의 표가 선택되지 않는 한 결합 상태는 필요하지 않습니다.

표 24. 내보낸 프롬프트 조회 오브젝트의 표 및 필드 번호

레코드 유형	표 번호	필드 번호	필드 설명
T	1110	-	표 정의 표
		1112	-- 표 ID(유효한 표 ID는 A-Z와 #, \$, @입니다.)
		1113	-- 표 이름
T	1150	-	결합 상태 표
		1152	-- 열 1 이름
		1153	-- 열 2 이름
T	1210	-	열 표
		1212	-- 열 유형: C=열 E=표현식 S=표현식이 있는 요약 기능 F=열만 있는 요약 기능
			1213
T	1310	-	행 선택 조건

표 24. 내보낸 프롬프트 조희 오브젝트의 표 및 필드 번호 (계속)

레코드 유형	표 번호	필드 번호	필드 설명
		1312	<p>-- 항목 유형</p> <p>1 - 연산자의 왼쪽</p> <p>2 - 연산자</p> <p>3 - 연산자의 오른쪽</p> <p>4 - 연결 연산자</p>
		1313	<p>-- 항목 유형 '1'의 경우, 열 유형을 식별합니다.</p> <p>C=열</p> <p>E=표현식</p> <p>S=요약 기능</p> <p>F=요약 기능(열 이름만 지정됩니다.)</p> <p>-- 항목 유형 '2'의 경우, 명령어를 식별합니다.</p> <p>'is'의 경우 IS(기본값)</p> <p>'is not'의 경우 ISN</p> <p>-- 항목 유형 '3'의 경우(사용되지 않음)</p> <p>-- 항목 유형 '4'의 경우, 연결 연산자를 식별합니다.</p> <p>'or'의 경우 O</p> <p>'and'의 경우 A(기본값)</p>
		1314	<p>-- 항목 유형 '1'의 경우 이 필드는 다음과 같습니다.</p> <p>열 이름, 표현식 또는 요약 기능</p> <p>-- 항목 유형 '2'의 경우, 연산자를 식별합니다.</p> <p>'같음'의 경우 EQ</p> <p>'보다 작음'의 경우 LT</p> <p>'이하'의 경우 LE</p> <p>'보다 큼'의 경우 GT</p> <p>'이상'의 경우 GE</p> <p>'사이값'의 경우 BT</p> <p>'시작값'의 경우 SW</p> <p>'종료값'의 경우 EW</p> <p>'포함'의 경우 CT</p> <p>NULL 경우의 NL</p> <p>-- 항목 유형 '3'의 경우, 값을 식별합니다.</p> <p>-- 항목 유형 '4'의 경우(사용되지 않음)</p>
T	1410	-	정렬 조건 표

내보내기/가져오기 형식

표 24. 내보낸 프롬프트 주희 오브젝트의 표 및 필드 번호 (계속)

레코드 유형	표 번호	필드 번호	필드 설명
V		1412	-- 정렬 방향 '오름차순'의 경우 A '내림차순'의 경우 D
		1413	-- 열
		1501	중복 행 처리 '유지'의 경우 K '버림'의 경우 D

1313 및 1314 필드 값의 의미는 1310 표 내의 필드 번호 1312에 지정된 순차 번호에 따라 다릅니다.

서식 오브젝트의 표 및 필드 번호

표25에서는 서식 오브젝트의 T 레코드의 표 번호 및 V 레코드의 필드 번호를 보여줍니다. 각 번호는 서식의 특정 부분과 일치합니다.

필드 3080인 V 레코드는 그 뒤에 오는 구분 패널에 대해 『트리거』로서 작동합니다. 이 레코드는 서식의 모든 구분 패널에 대해 한 번 나타납니다. 필드 값은 필드 3080 뒤에 오는 필드가 설명하는 구분 패널의 수를 반영합니다.

표 25. 내보낸 서식 오브젝트의 표 및 필드 번호

표 및 필드 번호	레코드 유형	설명	서식 패널
1110	T	열 표제 표	FORM.COLUMNS
1112	R	열 데이터 유형 ⁴	FORM.COLUMNS
1113	R	열 표제	FORM.COLUMNS
1114	R	열 사용 코드	FORM.COLUMNS
1115	R	열 들여쓰기	FORM.COLUMNS
1116	R	열 너비	FORM.COLUMNS
1117	R	열 편집 코드	FORM.COLUMNS
1118	R	열 순서	FORM.COLUMNS
1119	R	열 표제 배열 ⁵	FORM.COLUMNS
1120	R	열 데이터 배열 ⁵	FORM.COLUMNS

4. QMF는 서식 패널에 이 필드를 표시하지 않습니다.

5. 이 필드는 버전 3에 대해 새롭게 추가되었습니다.

표 25. 내보낸 서식 오브젝트의 표 및 필드 번호 (계속)

표 및 필드 번호	레코드 유형	설명	서식 패널
1121	R	열 정의 ⁵	FORM.COLUMNS
1122	R	열 정의 ⁵ 에 널(null) 패스	FORM.COLUMNS
1180	T	요약 계산 표	FORM.CALC
1182	R	계산 확인 번호	FORM.CALC
1183	R	요약 계산식	FORM.CALC
1184	R	요약 계산 너비	FORM.CALC
1185	R	요약 계산 편집 코드	FORM.CALC
1186	R	계산 ⁵ 에 널(null) 패스	FORM.CALC
1201	V	표제 앞의 공백행	FORM.PAGE
1202	V	표제 뒤의 공백행	FORM.PAGE
1210	T	페이지 표제 표	FORM.PAGE
1212	R	페이지 표제 행 번호	FORM.PAGE
1213	R	페이지 표제 배열	FORM.PAGE
1214	R	페이지 표제 텍스트	FORM.PAGE
1301	V	각주 앞의 공백행	FORM.PAGE
1302	V	각주 뒤의 공백행	FORM.PAGE
1310	T	페이지 각주 표	FORM.PAGE
1312	R	페이지 각주 행 번호	FORM.PAGE
1313	R	페이지 각주 배열	FORM.PAGE
1314	R	페이지 각주 텍스트	FORM.PAGE
1401	V	마지막 텍스트에 대한 새 페이지	FORM.FINAL
1402	V	최종 요약 행 번호	FORM.FINAL
1403	V	마지막 텍스트 앞의 공백행	FORM.FINAL
1410	T	마지막 텍스트 표	FORM.FINAL
1412	R	마지막 텍스트 행 번호	FORM.FINAL
1413	R	마지막 텍스트 배열	FORM.FINAL
1414	R	마지막 텍스트	FORM.FINAL
1501	V	세부 행 공간	FORM.OPTIONS
1502	V	구분 열 개요	FORM.OPTIONS
1503	V	기본 구분 텍스트	FORM.OPTIONS
1504	V	그룹화를 위한 열 표제에 있는 함수 이름	FORM.OPTIONS
1505	V	한 페이지에 자동 자동 준비된 열 유지	FORM.OPTIONS
1506	V	가로 요약 열	FORM.OPTIONS
1507	V	열 표제에 대한 분리자	FORM.OPTIONS
1508	V	구분 요약에 대한 분리자	FORM.OPTIONS
1509	V	가로 표제에 대한 분리자	FORM.OPTIONS
1510	V	최종 요약에 대한 분리자	FORM.OPTIONS
1511	V	자동 준비된 보고서 행의 너비	FORM.OPTIONS
1512	V	구분에서 페이지 번호 다시 매기기	FORM.OPTIONS

표 25. 내보낸 서식 오브젝트의 표 및 필드 번호 (계속)

표 및 필드 번호	레코드 유형	설명	서식 패널
1513	V	구분 또는 마지막 텍스트의 너비	FORM.OPTIONS
1514	V	열 재정렬	FORM.OPTION
1515	V	고정 열	FORM.OPTIONS
2790	V	세부 변이 번호	FORM.DETAIL
2791	V	세부 변이 선택	FORM.DETAIL
2805	V	열 표제 포함	FORM.DETAIL
2810	T	세부 표제 표	FORM.DETAIL
2812	R	세부 표제 텍스트 행	FORM.DETAIL
2813	R	세부 표제 배열	FORM.DETAIL
2814	R	세부 표제 텍스트	FORM.DETAIL
2901	V	세부 텍스트에 대한 새 페이지	FORM.DETAIL
2902	V	열 데이터의 행 번호	FORM.DETAIL
2904	V	세부 텍스트 뒤에 생략한 행의 수	FORM.DETAIL
2906	V	세부 표제 반복	FORM.DETAIL
2907	V	함께 보관할 세부 텍스트 행의 번호	FORM.DETAIL
2910	T	세부 텍스트 표	FORM.DETAIL
2912	R	세부 텍스트 행 번호	FORM.DETAIL
2913	R	세부 텍스트 배열	FORM.DETAIL
2914	R	세부 텍스트	FORM.DETAIL
3080	V	구분 패널 번호 ⁵	FORM.BREAKn
3101	V	구분 표제 ⁵ 용 새 페이지	FORM.BREAKn
3102	V	구분 표제 ⁵ 반복	FORM.BREAKn
3103	V	구분 표제 ⁵ 전에 생략할 행 번호	FORM.BREAKn
3104	V	구분 표제 ⁵ 뒤에 생략할 행 번호	FORM.BREAKn
3110	T	구분 표제 텍스트 표 ⁵	FORM.BREAKn
3112	R	구분 표제 행 번호 ⁵	FORM.BREAKn
3113	R	구분 표제 배열 ⁵	FORM.BREAKn
3114	R	구분 표제 텍스트 ⁵	FORM.BREAKn
3201	V	구분 텍스트 ⁵ 용 새 페이지	FORM.BREAKn
3202	V	구분 텍스트 요약 행 ⁵	FORM.BREAKn
3203	V	구분 텍스트 ⁵ 전에 생략할 행 번호	FORM.BREAKn
3204	V	구분 텍스트 ⁵ 뒤에 생략할 행 번호	FORM.BREAKn
3210	T	구분 텍스트 표 ⁵	FORM.BREAKn
3212	R	구분 텍스트 행 ⁵	FORM.BREAKn
3213	R	구분 텍스트 배열 ⁵	FORM.BREAKn
3214	R	구분 텍스트 ⁵	FORM.BREAKn
3310	T	조건 표 ⁵	FORM.CONDITIONS
3312	R	조건 확인 번호 ⁵	FORM.CONDITIONS
3313	R	조건부 표현식 ⁵	FORM.CONDITIONS
3314	R	조건 패널 ⁵ 에 널(null) 패스	FORM.CONDITIONS

표26에서는 버전 3 릴리스 1 전에 작성된 오브젝트에 대한 유효한 필드를 보여줍니다. QMF는 입력에서는 이 필드를 허용하지만, 출력에서는 이 필드를 작성하지 않습니다. 각 구분 패널마다 고유한 필드 번호 세트가 있습니다.

표 26. QMF 3.1 이전에 내보낸 서식 오브젝트의 필드 번호

표 및 필드 번호	레코드 유형	설명	서식 패널
1601	V	BREAK1: 표제용 새 페이지	FORM.BREAK1
1602	V	BREAK1: 열 표제 반복	FORM.BREAK1
1603	V	BREAK1: 표제 앞의 공백행	FORM.BREAK1
1604	V	BREAK1: 표제 뒤의 공백행	FORM.BREAK1
1610	T	BREAK1: 표제 표	FORM.BREAK1
1612	R	BREAK1: 표제 행	FORM.BREAK1
1612	R	BREAK1: 표제 배열	FORM.BREAK1
1614	R	BREAK1: 표제 텍스트	FORM.BREAK1
1701	V	BREAK1: 각주용 새 페이지	FORM.BREAK1
1702	V	BREAK1: 열 각주 반복	FORM.BREAK1
1703	V	BREAK1: 각주 앞의 공백행	FORM.BREAK1
1704	V	BREAK1: 각주 뒤의 공백행	FORM.BREAK1
1710	T	BREAK1: 각주 표	FORM.BREAK1
1712	R	BREAK1: 각주 행	FORM.BREAK1
1713	R	BREAK1: 각주 배열	FORM.BREAK1
1714	R	BREAK1: 각주 텍스트	FORM.BREAK1
1801	V	BREAK2: 표제용 새 페이지	FORM.BREAK2
1802	V	BREAK2: 열 표제 반복	FORM.BREAK2
1803	V	BREAK2: 표제 앞의 공백행	FORM.BREAK2
1804	V	BREAK2: 표제 뒤의 공백행	FORM.BREAK2
1810	T	BREAK2: 표제 표	FORM.BREAK2
1812	R	BREAK2: 표제 행	FORM.BREAK2
1813	R	BREAK2: 표제 배열	FORM.BREAK2
1814	R	BREAK2: 표제 텍스트	FORM.BREAK2
1901	V	BREAK2: 각주용 새 페이지	FORM.BREAK2
1902	V	BREAK2: 열 각주 반복	FORM.BREAK2
1903	V	BREAK2: 각주 앞의 공백행	FORM.BREAK2
1904	V	BREAK2: 각주 뒤의 공백행	FORM.BREAK2
1910	T	BREAK2: 각주 표	FORM.BREAK2
1912	R	BREAK2: 각주 행	FORM.BREAK2
1913	R	BREAK2: 각주 배열	FORM.BREAK2
1914	R	BREAK2: 각주 텍스트	FORM.BREAK2
2001	V	BREAK3: 표제용 새 페이지	FORM.BREAK3

내보내기/가져오기 형식

표 26. QMF 3.1 이전에 내보낸 서식 오브젝트의 필드 번호 (계속)

표 및 필드 번호	레코드 유형	설명	서식 패널
2002	V	BREAK3: 열 표제 반복	FORM.BREAK3
2003	V	BREAK3: 표제 앞의 공백행	FORM.BREAK3
2004	V	BREAK3: 표제 뒤의 공백행	FORM.BREAK3
2010	T	BREAK3: 표제 표	FORM.BREAK3
2012	R	BREAK3: 표제 행	FORM.BREAK3
2013	V	BREAK3: 표제 배열	FORM.BREAK3
2014	R	BREAK3: 표제 텍스트	FORM.BREAK3
2101	V	BREAK3: 각주용 새 페이지	FORM.BREAK3
2102	V	BREAK3: 열 각주 반복	FORM.BREAK3
2103	V	BREAK3: 각주 앞의 공백행	FORM.BREAK3
2104	V	BREAK3: 각주 뒤의 공백행	FORM.BREAK3
2110	T	BREAK3: 각주 표	FORM.BREAK3
2112	R	BREAK3: 각주 행	FORM.BREAK3
2113	R	BREAK3: 각주 배열	FORM.BREAK3
2114	R	BREAK3: 각주 텍스트	FORM.BREAK3
2201	V	BREAK4: 표제용 새 페이지	FORM.BREAK4
2202	V	BREAK4: 열 표제 반복	FORM.BREAK4
2203	V	BREAK4: 표제 앞의 공백행	FORM.BREAK4
2204	V	BREAK4: 표제 뒤의 공백행	FORM.BREAK4
2210	T	BREAK4: 표제 표	FORM.BREAK4
2212	R	BREAK4: 표제 행	FORM.BREAK4
2213	R	BREAK4: 표제 배열	FORM.BREAK4
2214	R	BREAK4: 표제 텍스트	FORM.BREAK4
2301	V	BREAK4: 각주용 새 페이지	FORM.BREAK4
2301	V	BREAK4: 열 각주 반복	FORM.BREAK4
2303	V	BREAK4: 각주 앞의 공백행	FORM.BREAK4
2304	V	BREAK4: 각주 뒤의 공백행	FORM.BREAK4
2310	T	BREAK4: 각주 표	FORM.BREAK4
2312	R	BREAK4: 각주 행	FORM.BREAK4
2313	R	BREAK4: 각주 배열	FORM.BREAK4
2314	R	BREAK4: 각주 텍스트	FORM.BREAK4
2401	V	BREAK5: 표제용 새 페이지	FORM.BREAK5
2402	V	BREAK5: 열 표제 반복	FORM.BREAK5
2403	V	BREAK5: 표제 앞의 공백행	FORM.BREAK5
2404	V	BREAK5: 표제 뒤의 공백행	FORM.BREAK5
2410	T	BREAK5: 표제 표	FORM.BREAK5
2412	R	BREAK5: 표제 행	FORM.BREAK5
2413	R	BREAK5: 표제 배열	FORM.BREAK5

표 26. QMF 3.1 이전에 내보낸 서식 오브젝트의 필드 번호 (계속)

표 및 필드 번호	레코드 유형	설명	서식 패널
2414	R	BREAK5: 표제 텍스트	FORM.BREAK5
2501	V	BREAK5: 각주용 새 페이지	FORM.BREAK5
2502	V	BREAK5: 열 각주 반복	FORM.BREAK5
2503	V	BREAK5: 각주 앞의 공백행	FORM.BREAK5
2504	V	BREAK5: 각주 뒤의 공백행	FORM.BREAK5
2510	T	BREAK5: 각주 표	FORM.BREAK5
2512	R	BREAK5: 각주 행	FORM.BREAK5
2513	R	BREAK5: 각주 배열	FORM.BREAK5
2514	R	BREAK5: 각주 텍스트	FORM.BREAK5
2601	V	BREAK6: 표제용 새 페이지	FORM.BREAK6
2602	V	BREAK6: 열 표제 반복	FORM.BREAK6
2603	V	BREAK6: 표제 앞의 공백행	FORM.BREAK6
2604	V	BREAK6: 표제 뒤의 공백행	FORM.BREAK6
2610	T	BREAK6: 표제 표	FORM.BREAK6
2612	R	BREAK6: 표제 행	FORM.BREAK6
2613	R	BREAK6: 표제 배열	FORM.BREAK6
2614	R	BREAK6: 표제 텍스트	FORM.BREAK6
2701	V	BREAK6: 각주용 새 페이지	FORM.BREAK6
2702	V	BREAK6: 열 각주 반복	FORM.BREAK6
2703	V	BREAK6: 각주 앞의 공백행	FORM.BREAK6
2704	V	BREAK6: 각주 뒤의 공백행	FORM.BREAK6
2710	T	BREAK6: 각주 표	FORM.BREAK6
2712	R	BREAK6: 각주 행	FORM.BREAK6
2713	R	BREAK6: 각주 배열	FORM.BREAK6
2714	R	BREAK6: 각주 텍스트	FORM.BREAK6

보고서 오브젝트의 표와 필드 번호

다음 그림에서는 T 레코드에 대한 표 번호 및 V 레코드에 대한 필드 번호를 보여줍니다.

표 27. 일반적 보고서 내보낸 보고서 오브젝트의 표 및 필드 번호

표 및 필드 번호	레코드 유형	설명
1001	V	프로파일 DECIMAL 옵션
1002	V	L 레코드 제어 영역 + 고정 영역의 길이

내보내기/가져오기 형식

표 27. 일반적 보고서. 내보낸 보고서 오브젝트의 표 및 필드 번호 (계속)

표 및 필드 번호	레코드 유형	설명
1010	T	형식화된 보고서 표
1012	T	보고서에서 형식화된 각 데이터 열의 경우: OMIT를 제외한 모든 코드 사용법의 경우
1013	T	데이터 형식이 지정된 편집 코드
1014	T	형식이 지정된 데이터를 포함하는 필드의 시작 위치(들여쓰기 영역 포함)
1015	T	형식이 지정된 데이터를 포함하는 필드의 시작 위치(들여쓰기 영역 제외)
1016	T	형식이 지정된 데이터를 포함하는 필드의 종료 위치
1017	T	형식이 지정된 열 값이 나타나는 논리 보고서 행 내에서 상대적인 실제 보고서 행의 수

보고서가 가로 유형 보고서이면 1014, 1015 및 1016 필드의 의미에 대해서는 표 28 다음의 주2를 참조하십시오.

표 28. 가로 보고서. 내보낸 보고서 오브젝트의 필드 번호

필드 번호	레코드 유형	설명
2001	V	가로 값의 형식이 지정될 때 사용된 편집 코드
2002	V	가로 그룹별 데이터 행의 수
2003	V	가로 요약 열이 존재하는지 나타냅니다
2010	T	가로 보고서 표
2012	T	각 가로 값의 경우: 형식화된 가로 값의 시작 위치(가로값은 열 표제 행에 나타납니다.)
2013	T	형식화된 가로 값의 종료 위치
2014	T	선행하는 들여쓰기 영역을 포함하여 이 가로 값과 연관된 보고서 열 세트의 시작 위치

주:

1. 보고서 행의 위치 1은 즉시 L 레코드 고정 영역 바로 다음에 옵니다.
2. 가로 보고서의 집계 열의 경우, 1014, 1015 및 1016 필드는 집계된 열의 가로값의 설정 내에서 필드의 시작 및 종료의 위치 관계를 설명합니다(표28의 2014 필드를 참조하십시오).

3. 각 보고서 표제(PAGE 또는 BREAK) 또는 각주(PAGE, BREAK 또는 FINAL)에서 텍스트 행의 R 레코드는 서식 기본값 수정사항을 포함하는 마지막 행까지 작성됩니다.
주어진 표제 또는 각주에 대한 필드 모두가 해당된 원래의 값을 가질 경우에도 적어도 하나 이상의 R 레코드가 각 표제 또는 각주에 대해 작성됩니다.
4. 최대 레코드 길이가 초과되어도 연속 레코드가 보고서 오브젝트에 대해 작성됩니다.

QMF 보고서에서 사용되는 HTML 태그

표29은 QMF가 월드 와이드 웹에 표시되는 보고서의 형식을 지정하기 위해 사용하는 HTML 태그 설정을 간단히 설명합니다. 이 태그는 각각 시작 태그와 종료 태그로 되어 있습니다. 종료 태그는 정방향 슬래시(/)로 시작하고, 모든 태그는 꺾쇠 괄호(<>)로 묶여 있습니다. 이 태그에 대한 자세한 설명은 HTML 3.0 문서를 참조하십시오.

표 29. HTML 보고서에 사용되는 HTML 3.0 태그

태그 설정	설명
<HTML></HTML>	HTML 문서로서 파일을 정의합니다.
<HEAD></HEAD>	이 태그는 문서 헤더의 경계를 표시합니다.
<TITLE></TITLE>	QMF는 이 태그 사이에 단어 "Report"를 삽입합니다. 이 태그 사이의 내용은 HTML 문서 제목으로 들어갑니다. 제목의 위치는 브라우저 및 플랫폼에 따라 달라집니다. 이 태그는 헤더 내부에 위치합니다.
<BODY></BODY>	이 태그는 헤더 다음에 오고 문서의 본문을 포함합니다. 보고서 출력은 문서의 본문에 위치합니다.
<PRE></PRE>	이 태그 사이의 내용은 그대로 표시됩니다. 이 사이에서는 HTML 형식화가 이루어지지 않습니다. QMF는 HTML 문서의 본문에서 이 태그 사이에 보고서 출력을 배치합니다.

내보내기/가져오기 형식

부록C. 통합 교환 형식(IXF)

DATAFORMAT=IXF 옵션을 사용하여 DATA 또는 TABLE 오브젝트를 내보내기 위해 EXPORT 명령을 사용하면 파일, 데이터 세트 또는 CICS 데이터 대기열을 통합 교환 형식(IXF)으로 내보냅니다. QMF는 IXF의 서브세트를 지원하며 이에 대해서는 이 절에서 설명합니다. 완전한 통합 교환 형식에 대한 설명은 *Data Extract: Reference*를 참조하십시오.

QMF 환경 외부에서 표를 작성하고 가져오려는 경우 IXF 형식이 특히 유용합니다. 이것을 실행하려면 OUTPUTMODE를 CHARACTER로 설정하십시오.

QMF에서 내보낸 IXF 파일, 데이터 세트 또는 CICS 데이터 대기열은 다음 레코드로 구성되어 있습니다.

- 헤더 레코드(H)
- 표 레코드(T)
- 열 레코드(C)
- 데이터 레코드(D)

내보낸 파일, 데이터 세트 또는 CICS 데이터 대기열은 하나의 T 레코드와 하나의 H 레코드로 되어 있습니다. T 레코드에는 T 레코드 다음에 오는 C 레코드의 수가 포함되어 있습니다. 표의 각 열마다 C 레코드가 있습니다. D 레코드가 C 레코드 다음에 옵니다. 표의 각 행마다 D 레코드가 있습니다. 내보낸 파일, 데이터 세트 또는 CICS 데이터 대기열에서 레코드의 배열은 그림69와 같습니다.

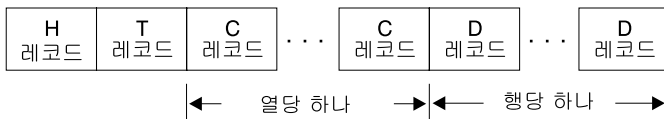


그림 69. 내보낸 데이터 파일, 데이터 세트 또는 CICS 데이터 대기열에서 레코드의 배열 (IXF 형식)

주: 데이터베이스 담당자 PC/IXF 파일 형식은 System/370™ IXF 형식과 동일하지 않습니다. IXF 형식의 데이터는 PC와 System/370 플랫폼 사이에서 전송될 수 없습니다.

통합 교환 형식(IXF)

다음 절은 각 레코드에 대한 형식을 설명합니다. 데이터를 내보냈을 경우 괄호 안에 표시되는 값은 MF가 제공한 값입니다.

헤더 레코드(H)

헤더 레코드(필수)는 파일, 데이터 세트 또는 CICS 데이터 대기열의 첫번째 레코드입니다. 그것은 문자 데이터를 포함하는 42바이트 레코드입니다. H 레코드의 형식은 다음과 같습니다.

바이트 위치	정보 및 유형
01	헤더 레코드 표시기(H)
02-04	파일, 데이터 세트 또는 CICS 데이터 대기열 ID(IXF)
05-08	IXF 버전(0000)
09-14	최초 제품 이름(QMF)
15-20	최초 제품 릴리스 레벨 (V7R1M0)
21-28	YYYYMMDD 서식으로 작성된 파일, 데이터 세트 또는 CICS 데이터 대기열의 날짜
29-34	HHMMSS 서식으로 작성된 파일, 데이터 세트 또는 CICS 데이터 대기열의 시간
35-39	파일, 데이터 세트 또는 CICS 데이터 대기열에서 첫번째 D(Data) 레코드를 선택하는 레코드 수. 이것은 문자 서식으로 표현되는 5 자리 숫자의 값입니다.
40	DBCS 표시기. DBCS 데이터가 가능한지를 Y 또는 N으로 알려줍니다.
41-42	예약됨

표 레코드(T)

표 레코드는 헤더 레코드 다음에 옵니다. 각 IXF 파일, 데이터 세트 또는 CICS 데이터 대기열은 T 레코드를 가집니다. 표 레코드는 파일, 데이터 세트 또는 CICS 데이터 대기열과 관계있는 표 및 데이터 정보를 포함합니다. T 레코드의 형식은 다음과 같습니다.

바이트 위치	정보 및 유형
01	표 레코드 표시기(T)
02-03	데이터 이름 길이(18)

바이트 위치	정보 및 유형
04-21	데이터가 검색되는 표의 이름으로 왼쪽 정렬되고 오른쪽은 공백으로 채워집니다. 표에 이름이 없으면 전체 18바이트 필드는 공백입니다.
22-29	데이터 이름 규정자. 데이터가 검색되는 데이터베이스 표의 소유자의 이름. 표에 소유자의 이름이 없으면 8바이트 필드는 공백입니다.
30-41	데이터 소스(데이터베이스)
42	데이터를 설명하는 데 사용된 규약. 열 데이터의 경우 C
43	데이터 형식: 문자의 경우 C(OUTPUTMODE=CHARACTER), 기계의 경우 M(OUTPUTMODE=BINARY)
44	데이터 위치: 내부의 경우 I
45-49	열(C) 레코드의 계수. 첫번째 데이터(D) 레코드 앞의 C 레코드의 수를 지정하는 문자 서식에서 숫자의 값
50-51	예약됨
52-81	공백

열 레코드(C)

열 레코드는 열의 데이터 특성을 설명합니다. 표의 각 열마다 열 레코드가 있습니다. 열 레코드의 형식은 다음과 같습니다.

바이트 위치	정보 및 유형
01	열 레코드 표시기(C)
02-03	열 이름 길이
04-21	데이터베이스로부터 획득되거나 QMF에 의해 생성된 열 이름(열에 원래 이름이 없었던 경우). 이름은 왼쪽 정렬되며 필요하면 오른쪽이 공백으로 채워집니다.
22	널(null)이 허용되면 알려주는 표시기(Y 또는 N)
23	선택된 열 표시기(Y)
24	키 열 표시기(Y)
25	데이터 클래스(R)
26-28	데이터 유형(데이터 유형 코드의 경우 124 페이지의 표12 참조)
29-33	코드 페이지(00000)
34-38	예약됨

통합 교환 형식(IXF)

바이트 위치	정보 및 유형
39-43	열 데이터 길이. 문자 서식에서 숫자의 값. 데이터 유형이 DECIMAL이면 첫번째 3바이트는 데이터 정밀도를 나타내고 다음 2바이트는 배율을 나타냅니다. 데이터 유형이 INTEGER, SMALLINT, DATE, TIME 또는 TIMESTAMP이면 이 필드는 공백입니다(길이는 데이터 유형에 따라 고유합니다).
44-49	열 데이터의 시작 위치. 값은(문자 서식에서) 데이터 레코드의 시작으로부터 열에 대한 데이터의 오프셋을 반영합니다. 열이 널(null)을 허용하면 이 필드는 널(null) 표시기를 가리킵니다. 열이 널(null)을 허용하지 않으면 데이터 자체를 가리킵니다. 열이 널(null)을 허용하는지 여부에 관계없이 널(null) 표시기에 대한 공간은 항상 레코드에 존재합니다. 시작 위치는 데이터를 포함하는 첫번째 바이트를 기준으로 합니다. 그러므로 데이터(D) 레코드의 첫번째 다섯 바이트는 시작 위치의 어떠한 고려사항에서도 포함되지 않습니다. 첫번째 데이터 위치는 위치 1이고 위치 0이 아닙니다.
50-79	열 레이블 정보(사용할 수 있는 경우). 공백(사용할 수 없을 경우)
80-81	문자 서식에서 두 바이트의 0(00)

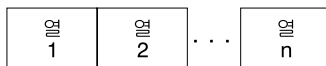
데이터 레코드(D)

표의 각 행마다 데이터 레코드가 있습니다. 데이터 레코드의 형식은 다음과 같습니다.

바이트 위치	정보 및 유형
01	데이터 레코드 표시기(D)
02-04	예약됨
05	공백
06-레코드의 끝	표 레코드의 바이트 43이 M(machine)인지 C(character)인지에 따라 2진 또는 문자 서식의 행 데이터. 바이트 6은 첫번째 열의 행 데이터의 시작(위치 1)을 나타냅니다.

열 데이터 형식

n 열의 D 레코드에서 데이터는 나란히 배치됩니다.



각 열의 경우, 데이터는 데이터 자체와 그 다음의 널(null) 표시기로 구성됩니다. 널(null)이 허용되면(C 레코드의 22바이트 = Y), 각 C 레코드의 바이트 44-49는 해당 열이 데이터 앞에 오는 널(null) 표시기를 가리킵니다. 바이트 22 = N(널(null)이 허용되지 않음)이면, 바이트 44-49는 데이터 자체를 가리킵니다. 그러나, 후자의 경우 널(null) 표시기용 공간은 데이터 레코드에 남아 있습니다. 바이트 44-49의 첫번째 위치는 1값으로 표시되며, D 레코드의 바이트 6을 가리킵니다(바이트 1-5는 무시됨).

널(null) 표시기의 표현은 OUTPUTMODE에 대해 문자 또는 2진수 중 어떤 값이 지정되었느냐에 따라 달라집니다. OUTPUTMODE는 T 레코드의 바이트 43(문자의 경우 C, 기계의 경우 M(2진))에 반영됩니다. 데이터 형식이 문자이면, 널(null) 표시기에 대해 1바이트가 사용됩니다.

- (대시)는 데이터가 널(null)임을 나타냅니다.
- (공백)은 데이터가 널(null)이 아님을 나타냅니다.

한 경우에는 널(null)이고 다른 경우에는 널(null)이 아닌 데이터를 보여주는 두 개의 D 레코드의 실례를 보려면 278 페이지의 그림70을 참조하십시오.

데이터 형식이 2진인 경우 널(null) 표시기에 대해 두 바이트가 사용됩니다.

- X'FFFF'는 데이터가 널(null)임을 나타냅니다.
- X'0000'은 데이터가 널(null)이 아님을 나타냅니다.

280 페이지의 그림71에서는 널(null)과 널(null)이 아닌 데이터 표시기 모두를 설명하는 두 개의 D 레코드를 보여줍니다.

데이터 유형별 열 데이터 형식

270 페이지의 표30에서는 문자와 2진 형식 모두에 대해 각 데이터 유형별 D 레코드 열 데이터의 길이 및 형식을 보여줍니다. 표에서, IXFCLENG는 C 레코드의 바이트 39-43의 내용(열 데이터의 길이)을 나타냅니다.

통합 교환 형식(IXF)

표 30. 데이터 유형별 IXF 열 데이터의 형식

데이터 유형 코드	데이터 유형	데이터 길이 정보 문자 형식	데이터 길이 정보 2진 형식
384	DATE	<p>IXFCLENG에서 값은 유효하지 않습니다. 길이(10바이트)는 데이터 유형에 따라 고유합니다.</p> <p>형식은 다음과 같습니다.</p> <p style="text-align: center;">yyyy-mm-dd</p> <p>여기서, yyyy는 년을, mm은 월을, dd는 일을 나타냅니다. yyyy, mm 및 dd는 숫자여야 합니다. 맨 앞 0은 생략될 수 없습니다. yyyy의 허용가능한 범위는 0001-9999이며, mm은 01-12입니다. dd 범위는 달에 따라 다릅니다. 예제:</p>	문자 형식과 같음

표 30. 데이터 유형별 IXF 열 데이터의 형식 (계속)

데이터 유형 코드	데이터 유형	데이터 길이 정보 문자 형식	데이터 길이 정보 2진 형식
388	TIME	<p>IXFCLENG에서 값은 유효하지 않습니다. 길이(8바이트)는 데이터 유형에 따라 고유합니다.</p> <p>형식은 다음과 같습니다.</p> <p style="text-align: center;">hh.mm.ss</p> <p>여기서, hh는 24시간 형식의 시간을, mm은 분을, ss는 초를 나타냅니다. hh, mm 및 ss는 모두 숫자여야 합니다. 맨 앞 0은 생략될 수 없습니다. 허용되는 범위는 다음과 같습니다.</p> <p style="text-align: center;">hh의 경우 00 - 23 mm의 경우 00 - 59 ss의 경우 00 - 59</p> <p>자정에 대한 특수값 24.00.00은 유효합니다. 예제:</p> <p>10.37.42는 10:37:42 AM 08.00.00은 정각 8 AM 23.30.00은 11:30 PM</p>	문자 형식과 같음

통합 교환 형식(IXF)

표 30. 데이터 유형별 IXF 열 데이터의 형식 (계속)

데이터 유형 코드	데이터 유형	데이터 길이 정보 문자 형식	데이터 길이 정보 2진 형식
392	TIMESTAMP	<p>IXFCLENG에서 값은 유효하지 않습니다. 길이(26바이트)는 데이터 유형에 따라 고유합니다.</p> <p>형식은 다음과 같습니다.</p> <pre>yyyy-mm-dd-hh .mm.ss.nnnnnn</pre> <p>여기서, yyyy는 년, 첫번째 mm은 월, dd는 일이며, hh는 24시간 형식의 시간, 두 번째 mm은 분, ss는 초이며, nnnnnn은 마이크로초입니다. 년, 월, 일, 시간, 분 및 초에 대한 유효한 범위는 DATE 및 TIME 데이터 유형과 같습니다. nnnnnn은 000000-999999일 수 있습니다. 예제:</p> <pre>1997-12-31-23 .59.59.999999 (1997년의 마지막 마이크로초) 1998-01-01-00 .00.00.000000 (1998년의 첫번째 마이크로초)</pre> <p>24.00.00.000000은 시간소인의 시간 부분에 대해 유효합니다.</p>	문자 형식과 같음

표 30. 데이터 유형별 IXF 열 데이터의 형식 (계속)

데이터 유형 코드	데이터 유형	데이터 길이 정보 문자 형식	데이터 길이 정보 2진 형식
448	VARCHAR	<p>IXFLENGTH는 문자열의 최대 길이입니다. 데이터 길이는 5바이트 문자 계수 필드가 선행하는 IXFLENGTH에 의해 표시되는 N 바이트로 구성됩니다(N에 허용 가능한 범위는 0-254이며, 계수 필드의 경우에는 0-N입니다). 계수 필드에 의해 표시되는 문자의 수는 유효합니다. 나머지는 무의미합니다. 예제:</p> <p>IXFLENGTH=00010이면 데이터 형식은 00005JONESxxxxx입니다.</p> <p>여기서, 각 x는 공백 문자(X'40')입니다.</p>	<p>IXFLENGTH는 문자열의 최대 길이입니다. 데이터 길이는 2바이트 2진 계수 필드가 선행하는 IXFLENGTH에 의해 표시되는 N 바이트로 구성됩니다(N에 허용 가능한 범위는 1-254이며 계수 필드의 경우에는 0-N입니다). 계수 필드에 의해 표시되는 문자의 수는 유효합니다. 나머지는 무의미합니다. 예제:</p> <p>IXFLENGTH=00010이면 데이터 형식은 nnJONESxxxxx입니다.</p> <p>여기서, nn=X'0005' 및 각 x는 공백 문자(X'40')입니다.</p>
452	CHAR	<p>IXFLENGTH는 문자열의 길이입니다. 데이터 길이는 IXFLENGTH의 N 바이트에 의해 표시됩니다(N에 허용 가능한 범위는 1-254입니다). 예제:</p> <p>IXFLENGTH=00005이면 데이터 형식은 JONES입니다.</p> <p>여기서, JONES는 C 레코드의 바이트 44-49로 지정되는 5바이트의 문자열입니다.</p>	문자 형식과 같음
456	LONG VARCHAR	N에 허용 가능한 범위가 0-32767이라는 점을 제외하고는 VARCHAR과 같습니다.	문자 형식과 같음

통합 교환 형식(IXF)

표 30. 데이터 유형별 IXF 열 데이터의 형식 (계속)

데이터 유형 코드	데이터 유형	데이터 길이 정보 문자 형식	데이터 길이 정보 2진 형식
464	VARGRAPHIC	<p>IXFCLENG는 2바이트 문자의 최대 수입니다(2xN 바이트). 데이터 길이는 5바이트 문자 계수 필드, IXFCLENG가 가리키는 바이트 수의 2배수, 2(시프트 문자용)로 구성됩니다. 계수 필드에서 2바이트 문자의 수는 데이터 바로 앞의 SO(X'0E')와 데이터 바로 뒤의 SI(X'0F')가 합쳐져서 유효해집니다. 나머지는 무의미하게 됩니다(N에 허용 가능한 범위는 1-127이며, 계수 필드의 경우에는 0-N입니다). 예제:</p> <p>IXFCLENG = 00006이면 데이터 형식은 00003oZZYXXixxxxx입니다.</p> <p>여기서, o는 SI이고 i는 SI이며 각 x는 공백 문자(X'40')입니다.</p>	<p>데이터 길이는 2바이트 2진 계수 필드와 IXFCLENG에 의해 표시되는 바이트의 2배수로 구성됩니다. IXFCLENG의 허용가능한 범위는 1-127이며, 계수 필드의 경우에는 0-IXFCLENG입니다. 계수 필드에서 2바이트 문자의 수는 유효합니다. 어느 것도 SO와 SI 문자로 묶이지 않습니다. 나머지는 무의미하게 됩니다. 예제:</p> <p>IXFCLENG = 00008이면 데이터 형식은 nnZZYXXWwxxxxxx입니다.</p> <p>여기서, nn=X'0004'와 각 x는 공백 문자(X'40')입니다.</p>
468	GRAPHIC	<p>IXFCLENG는 2바이트 문자의 수(2*N 바이트)입니다. 데이터 길이는 데이터 바로 앞에 있는 SO(X'0E')와 데이터 바로 뒤에 오는 SI(X'0F')와 2*N 바이트를 합한 수입니다. 예제:</p> <p>IXFCLENG=00005이면 데이터 형식은 oZZYXXWwVVi입니다.</p> <p>여기서, o는 SO이고 i는 SI입니다.</p>	<p>데이터 문자열에 둘러싸고 있는 SI 및 SO 문자가 없다는 것을 제외하고는 문자 형식과 같습니다. 예제:</p> <p>IXFCLENG=00005이면 데이터 형식은 ZZYXXWwVVi입니다.</p>
472	LONG VARGRAPHIC	<p>N에 허용가능한 범위가 0-16383이라는 점을 제외하고는 VARGRAPHIC과 같습니다.</p>	<p>문자 형식과 같음</p>

표 30. 데이터 유형별 IXF 열 데이터의 형식 (계속)

데이터 유형 코드	데이터 유형	데이터 길이 정보 문자 형식	데이터 길이 정보 2진 형식
480	FLOAT	<p>IXFLENG에서 값은 8입니다. 데이터의 길이 및 형식은 데이터 유형별로 고유합니다.</p> <p>데이터는 다음과 같이 배열된 23 바이트 문자 값으로 구성되어 있습니다.</p> <p>부호용 1 문자 가수용 18 문자(17 자리수 및 소수점) 문자 E 부호 있는 3 문자 지수</p> <p>예제: -1.2345678901234567E+14 +6.2345678901234567E-01 0.0000000000000000E+00</p>	<p>IXFLENG에서 값은 8입니다. 데이터의 길이 및 형식은 데이터 유형별로 고유합니다.</p> <p>데이터는 긴 부동 소수점에 대해 표준 IBM S/370™ 형식으로 된 8바이트 부동 소수점 값으로 구성되어 있습니다.</p>
484	DECIMAL	<p>C 레코드의 바이트 39-43은 정밀도 P(첫 3바이트) 및 배율 S(다음 2바이트)를 나타냅니다. P에 허용가능한 범위는 0 - 15입니다. S는 P보다 작거나 같은 값이 됩니다.</p> <p>데이터가 P+2 바이트 문자 값(또는 S=0인 경우 P+1 바이트)으로 형식화되고 오른쪽으로 정렬되며, 첫 바이트는 부호용으로 예약되어 있으므로 소수점(S에 의해 위치 지정)은 S가 0이 아닌 경우에만 표시됩니다. 예제: P=005이고 S=00이면, 데이터 형식은 12345입니다. P=006이고 S=02이면, 데이터 형식은 +2345.10입니다. P=004이고 S=03이면, 데이터 형식은 -8.515입니다.</p>	<p>C 레코드의 바이트 39-43은 정밀도 P(첫 3바이트) 및 배율 S(다음 2바이트)를 나타냅니다. P에 허용가능한 범위는 0 - 15입니다. S는 P보다 작거나 같은 값이 됩니다.</p> <p>데이터는 표준 IBM S/270 팩 십진 형식으로 된 (P+2)/2 바이트 팩 십진 값으로 구성되며, P 자리수의 S는 다음의 포함된 소수점으로 해석됩니다. 예제: P=005이고 S=00이면, 데이터 형식은 X'12345C'입니다. P=006이고 S=02이면, 데이터 형식은 X'0234510D'입니다.</p>

통합 교환 형식(IXF)

표 30. 데이터 유형별 IXF 열 데이터의 형식 (계속)

데이터 유형 코드	데이터 유형	데이터 길이 정보 문자 형식	데이터 길이 정보 2진 형식
496	INTEGER	IXFLENGTH에서 값은 유효하지 않습니다. 데이터의 길이 및 형식은 데이터 유형에 따라 고유합니다. 데이터는 오른쪽 정렬된 11바이트 문자 값으로 되어 있고 첫 번째 문자가 부호용으로 예약되어 있습니다. 예제: 0000000013 +1187642200 -0033588727	IXFLENGTH에서 값은 유효하지 않습니다. 데이터의 길이 및 형식은 데이터 유형에 따라 고유합니다. 데이터는 4바이트 2진 값으로 되어 있습니다.
500	SMALLINT	IXFLENGTH에서 값은 유효하지 않습니다. 데이터의 길이 및 형식은 데이터 유형에 따라 고유합니다. 데이터는 오른쪽 정렬된 6바이트 문자 값으로 구성되어 있고, 첫 번째 문자가 부호용으로 예약되어 있습니다. 예제: 00023 +00763 -21311	IXFLENGTH에서 값은 유효하지 않습니다. 데이터의 길이 및 형식은 데이터 유형에 따라 고유합니다. 데이터는 2바이트 2진 값으로 되어 있습니다.

IXF의 예제

93페이지의 예제에서처럼 표(QMF 형식을 사용하여 내보낸)를 이제 IXF 형식을 사용하여 내보냈다고 가정해 보십시오(OUTPUTMODE=CHARACTER로).

ID	NAME	COMM
10	SANDERS	-
20	PERNAL	612.45

내보낸 파일, 데이터 세트 또는 CICS 데이터 대기열은 총 7개의 레코드(하나의 H 레코드, 하나의 T 레코드, 세 개의 C 레코드 및 두 개의 D 레코드)로 구성되어 있습니다.

```

HIXF0000QMF  V7R1M01998120409560000005N
T18                database                CCI00003
C02ID              NYNR50000000           000002
C04NAME            YYNR44800000           00009000008  00
C04COMM            YYNR48400000           00702000023  00
D      00010 00007SANDERSxx-
D      00020 00006PERNALxxx 00612.45
    
```

인쇄할 수 없는 2진 문자는 x'로서 표시됩니다. 278 페이지의 그림70은 이 레코드에 관한 자세한 정보를 줍니다.

통합 교환 형식(IXF)

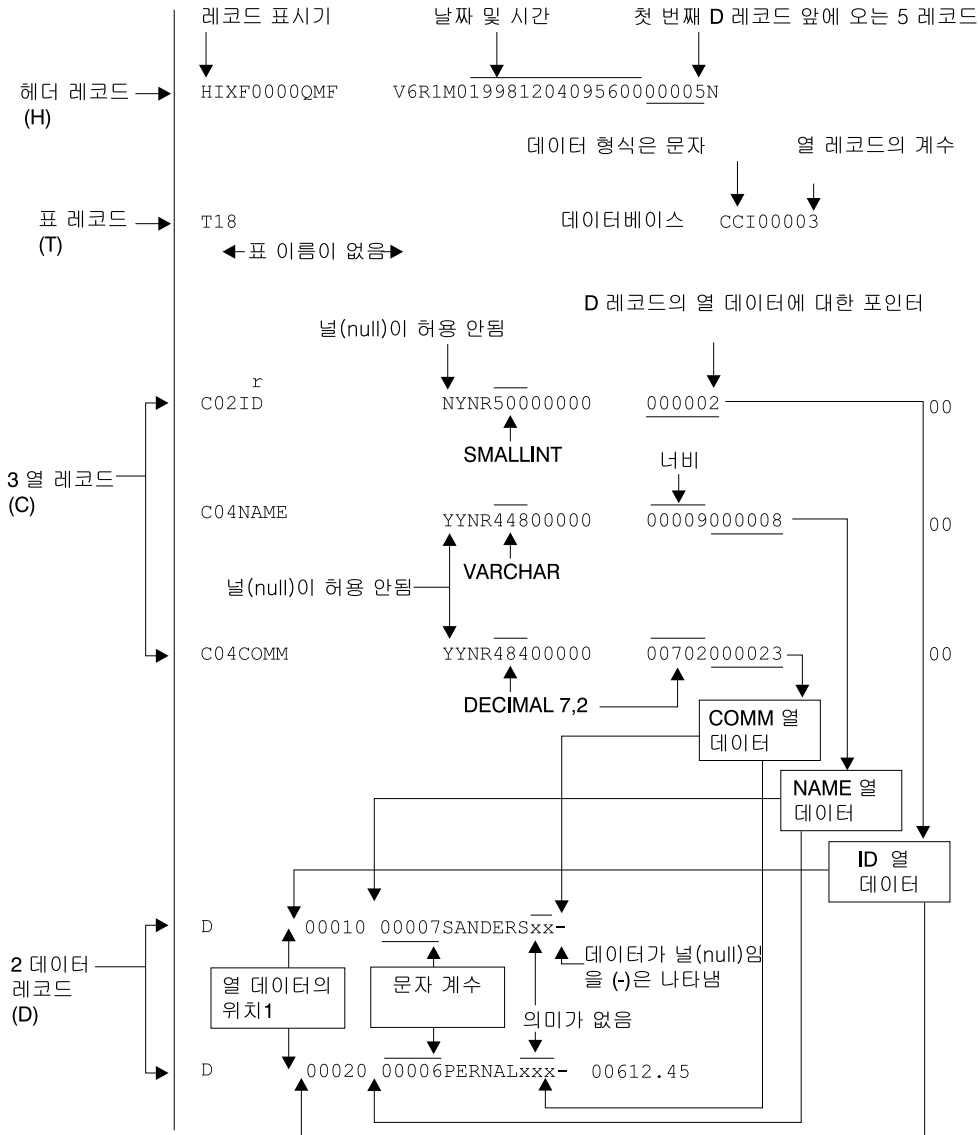


그림 70. 샘플 IXF 레코드의 형식(OUTPUTMODE=CHARACTER)

이제 같은 표를 IXF 형식을 사용하면서 OUTPUTMODE=BINARY로 내보냈다고 가정해 보십시오. 앞의 예제와 같이 내보낸 파일, 데이터 세트 또는 CICS 데이터 대기열은 다음과 같이 7개의 레코드로 되어 있습니다.

```
HIXF0000QMF  V7R1M01998120409565000005N
T18          database          CMI00003
C02ID       NYNR50000000      000003
```

00

C04NAME	YYNR44800000	00009000005	00
C04COMM	YYNR48400000	00702000018	00
D	xxxxxxxxSANDERSxxxxxxxx		
D	xxxxxxxxPERNALxxxxxxxx		

바이트 44에서 49까지(열 데이터의 시작 위치)를 제외하고 H, T, C 레코드의 정보는 본질적으로 같습니다. 그러나 D 레코드의 데이터는 상당히 다릅니다. 280 페이지의 그림 71에서는 내보낸 파일, 데이터 세트 또는 CICS 데이터 대기열의 레코드에 관한 자세한 정보를 보여줍니다.

부록D. ADDRESS QRW: QMF 명령 환경 사용

CICS 사용자를 위한 주의사항

REXX는 QMF CICS에 지원되지 않으므로, ADDRESS QRW도 CICS 환경에서 작동하지 않습니다.

REXX 언어는 항상 명령의 기본 분석을 결정하는 명령 환경에서 작동합니다. 기본 명령 환경은 CMS 또는 TSO로 운영 체제에 따라 달라집니다.

QMF가 시작되면 REXX ADDRESS 명령을 통해 기본 명령 환경으로 QMF를 설정할 수 있습니다. 이 명령을 단독으로 사용하거나 QMF 명령 앞에 사용할 수 있습니다.

```
ADDRESS QRW
```

```
ADDRESS QRW command
```

ADDRESS QRW가 설정되면 QMF는 다른 ADDRESS 명령을 발행할 때까지 기본 명령 환경을 유지합니다. ADDRESS QRW *command*가 설정되면 QMF는 해당 명령에 대해서만 명령 환경이 됩니다.

QMF 논리를 갖는 프로시저를 사용할 경우 QRW는 기본 명령 환경이 됩니다.

호출 인터페이스를 사용하든지 REXX 명령 환경을 사용하는지에 관계없이 QMF는 똑같이 작동하지만, ADDRESS QRW는 SAA 조회 CPI의 부분은 아닙니다. 응용프로그램을 다른 SAA 조회 환경으로 이식할 플랜이 없을 경우에만 이 명령을 사용합니다.

다음 예제에서는 QMF 명령 환경을 사용하는 방법을 보여줍니다.

```

:
:
call dsqcix "START (DSQSMODE=INTERACTIVE"
if dsq_return_code=dsq_severe | dsq_return_code=dsq_failure
  then exit dsq_return_code

ADDRESS QRW
"RUN PROC MONDAY_P"
if dsq_return_code=dsq_severe | dsq_return_code=dsq_failure
  then exit dsq_return_code

"EXIT"
if dsq_return_code=dsq_severe | dsq_return_code=dsq_failure
  then exit dsq_return_code
:
:

```

그림 72. QMF 명령 환경을 사용하는 예제

부록E. 제품 인터페이스 매크로

QMF는 이 부록에서 확인된 매크로를 고객에 대한 일반 사용자 프로그래밍 인터페이스로서 제공합니다.

경고: 이 부록에서 확인된 것 외에 어떠한 QMF 매크로도 프로그래밍 인터페이스로서 사용하지 마십시오.

제품 인터페이스 매크로

DSQQMF n

여기서, n 은 NLF ID입니다. 영어의 경우, 이 ID는 E입니다.

호출 인터페이스 매크로

어셈블러

- DSQCIA

- DSQCOMMA

C/370

- DSQCIC

- DSQCICE

- DSQCOMMC

COBOL

- DSQCIB

- DSQCOMMB

FORTRAN

- DSQCIF

- DSQCIFE

- DSQCOMMF

PL/I

- DSQCIPL

- DSQCIPX

- DSQCOMML

REXX

- DSQCIX

명령 인터페이스 매크로

DSQCCI

QMF 자원 관리자 종료 인터페이스 매크로

DXEGOVA

제품 인터페이스 매크로

DXEXCBA

QMF 사용자 편집 종료 매크로

DXEECS

부록F. QMF 글로벌 변수 표

QMF는 응용프로그램에서 사용할 많은 변수를 제공합니다. 버전 3에서 QMF는 호출 인터페이스에 대한 현재 명명 규약을 소개했습니다. 해당 명령 인터페이스 변수 이름은 여전히 유효합니다.

호출 인터페이스 글로벌 변수 이름은 최대 18자가 될 수 있습니다. 호출 인터페이스 사용자는 이전 이름(8자) 또는 새 이름(18자) 중 하나를 사용할 수 있습니다. 그러나 새 이름을 사용하는 것이 좋습니다. 명령 인터페이스 사용자는 이전 이름을 사용해야 합니다.

새 명명 규약은 **DSQcc_XXXXXXXXXXXX**입니다.

cc 다음의 범주 ID 중 하나가 될 수 있습니다.

- AP** 프로파일 관련 상태 정보
- AO** 다른(프로파일과 관련 없는) 상태 정보
- CM** 이전 명령에 의해 작성된 메시지에 대한 정보
- CP** 표 편집기에 관한 정보
- DC** QMF가 화면에 정보를 표시하는 방법을 제어합니다.
- EC** QMF가 명령 및 프로시저를 실행하는 방법을 제어합니다.
- QC** CONVERT QUERY 옵션에 의해 생성된 변수
- QM** RUN QUERY 오류 메시지 정보
- QW** Windows용 QMF에 고유한 변수

_ 밑줄 문자

XXXXXXXXXXXX

최대 12자의 설명적 이름

QMF는 버전 3.3에서부터 초기화시 글로벌 변수 조절을 허용하는 Q.SYSTEM_INI 라는 특수 프로시저를 제공합니다. 자세한 내용은 사용하는 운영 체제용 QMF *Installing and Managing*을 참조하십시오.

프로파일 관련 상태 정보에 대한 DSQ 글로벌 변수

다음 글로벌 변수는 SET GLOBAL 명령에 의해 변경될 수 없습니다.

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAP_CASE	DSQAPCAS	01	CASE 매개변수. 값은 다음과 같습니다. 1 UPPER의 경우 2 MIXED의 경우 3 STRING의 경우
DSQAP_CONFIRM	DSQAPRMP	01	CONFIRM 매개변수. 값은 다음과 같습니다. 0 NO의 경우 1 YES의 경우
DSQAP_DECIMAL	DSQAPDEC	01	DECIMAL 매개변수. 값은 다음과 같습니다. 1 PERIOD의 경우 2 COMMA의 경우 3 FRENCH의 경우
DSQAP_LENGTH	DSQAPLEN	18	LENGTH 매개변수. 해당되는 값은 매개변수의 값입니다('1'에서 '999' 또는 'CONT').
DSQAP_PFKEY_TABLE	DSQAPPFK	31	기능 키 표의 이름
DSQAP_PRINTER	DSQAPPRT	08	PRINTER 매개변수. 값은 다음과 같습니다. GDDM 프린터의 별명 DSQPRINT와 연관된 프린터용 공백

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAP_QUERY_LANG	DSQAPLNG	01	LANGUAGE 매개변수. 값은 다음과 같습니다. 1 SQL의 경우 2 QBE의 경우 3 PROMPTED의 경우
DSQAP_QUERY_MODEL	DSQAMODP	01	MODEL 매개변수. 값은 RELATIONAL의 경우 '1'입니다.
DSQAP_RESOURCE_GRP	DSQAPGRP	16	RESOURCE 매개변수
DSQAP_SPACE	DSQAPSPC	50	SPACE 매개변수. 해당되는 값은 매개변수의 값입니다
DSQAP_SYNONYM_TBL	DSQAPSYN	31	SYNONYMS 매개변수
DSQAP_TRACE	DSQAPTRC	18	TRACE 매개변수. 값은 다음과 같습니다. ALL (최대 추적) NONE (최소 추적) 개별 QMF 구성요소의 세부사항(예: A2L2C1)
DSQAP_WIDTH	DSQAPWID	18	WIDTH 매개변수. 해당되는 값은 매개변수의 값입니다('22'에서 '999').

프로파일과 관련없는 상태 정보에 대한 DSQ 글로벌 변수

다음 글로벌 변수는 SET GLOBAL 명령에 의해 변경될 수 없습니다.

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAO_APPL_TRACE	DSQATRAC	01	응용프로그램 추적 레벨. 값은 다음과 같습니다. 0 레벨 A0의 경우 1 레벨 A1의 경우 2 레벨 A2의 경우
DSQAO_ATTENTION	DSQCATTN	01	사용자 주의 플래그

QMF 글로벌 변수

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAO_BATCH	DSQABATC	01	일괄처리 또는 대화식 모드. 값은 다음과 같습니다. <ol style="list-style-type: none"> 1 대화식 세션의 경우 2 일괄처리 모드 세션의 경우
DSQAO_CONNECT_ID	DSQAAUTH	08	데이터베이스에 연결하는 데 사용되는 사용자 ID(이 ID를 통해 작업이 수행됩니다.)
DSQAO_CONNECT_LOC	없음	18	사용자가 현재 연결한 데이터베이스의 위치 이름. 이름은 18자입니다(필요하면, 오른쪽을 공백으로 채웁니다).
DSQAO_CURSOR_OPEN	DSQACRSR	01	데이터베이스 커서 상태. 값은 다음과 같습니다. <ol style="list-style-type: none"> 1 커서가 열려 있을 경우 2 커서가 닫혀 있을 경우
DSQAO_DB_MANAGER	DSQADBMG	01	데이터베이스 관리자. 값은 다음과 같습니다. <ol style="list-style-type: none"> 1 VM/ESA 또는 VSE/ESA용 DB2의 경우 2 MVS/ESA용 DB2의 경우 3 워크스테이션 데이터베이스 서버의 경우
DSQAO_DBCS	DSQADBCS	01	DBCS 지원 상태. 값은 다음과 같습니다. <ol style="list-style-type: none"> 1 DBCS 지원의 경우 2 DBCS 지원이 없는 경우

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAO_FORM_PANEL	DSQASUBP	02	<p>현재 서식 패널. 값은 다음과 같습니다.</p> <p>1 FORM.MAIN의 경우</p> <p>2 FORM.COLUMN의 경우</p> <p>3 FORM.PAGE의 경우</p> <p>4 FORM.FINAL의 경우</p> <p>5 FORM.BREAK1의 경우</p> <p>6 FORM.BREAK2의 경우</p> <p>7 FORM.BREAK3의 경우</p> <p>8 FORM.BREAK4의 경우</p> <p>9 FORM.BREAK5의 경우</p> <p>10 FORM.BREAK6의 경우</p> <p>11 FORM.OPTIONS의 경우</p> <p>12 FORM.CALC의 경우</p> <p>13 FORM.DETAIL의 경우</p> <p>14 FORM.CONDITIONS의 경우</p> <p>공백 값은 서식이 QMF 임시 기억영역에 존재하지 않는다는 의미입니다.</p>
DSQAO_INTERACT	DSQAIACT	01	<p>대화 플러그의 설정. 값은 다음과 같습니다.</p> <p>0 대화식 실행이 없는 경우</p> <p>1 대화식 실행이 허용되는 경우</p>

QMF 글로벌 변수

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAO_LOCAL_DB2	없음	18	지역 DB2 데이터베이스의 위치 이름. 이 이름은 변수 DSQAO_SUBSYS_ID에서 명명된 서브시스템의 위치 이름입니다. 원격 작업 단위 환경에서, DSQ_LOCAL_DB2는 응용프로그램 리퀘스터의 이름입니다. 이름은 16자입니다(필요하면, 오른쪽을 공백으로 채웁니다). QMF가 VM 또는 VSE 환경에서 실행될 경우 이 필드는 공백입니다.
DSQAO_LOCATION	DSQAITLO	16	현재 오브젝트의 위치 이름(있을 경우). 3가지 이름이 사용될 때에만 이 값을 적용할 수 있습니다.
DSQAO_NLF_LANG	DSQALANG	01	사용자의 자국어. 영어 환경의 경우 'E'입니다.
DSQAO_NUM_FETCHED	DSQAROWS	16	페치된 데이터 행. DATA 오브젝트가 공백일 경우 '0'을 포함합니다.
DSQAO_OBJ_NAME	DSQAITMN	18	현재 패널에 표시된 표 이름(보고서에 포함됨), 조회, 프로시저 또는 서식. 현재의 패널이 오브젝트를 보여주지 않거나 표시된 오브젝트에 이름이 없으면 이 변수는 공백을 포함합니다.
DSQAO_OBJ_OWNER	DSQAITMO	08	현재 패널에 표시된 표의 소유자(보고서에 포함됨), 조회, 프로시저 또는 서식. 현재의 패널이 오브젝트를 보여주지 않거나 표시된 오브젝트에 소유자가 없으면 이 변수는 공백을 포함합니다.

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAO_PANEL_TYPE	DSQAITEM	01	현재 패널의 유형. 값은 다음과 같습니다. 1 HOME의 경우 2 QUERY의 경우 3 REPORT의 경우 4 FORM의 경우 5 PROC의 경우 6 PROFILE의 경우 7 CHART의 경우 8 LIST의 경우 9 표 편집기의 경우 A GLOBALS의 경우
DSQAO_QMF_RELEASE	DSQAREVN	02	QMF의 숫자 릴리스 번호. QMF 버전 7의 경우, 이 번호는 '12'입니다.
DSQAO_QMF_VER_RLS	DSQAQMF	10	QMF의 버전 및 릴리스 QMF 버전 7의 경우 'QMF V7'입니다.
DSQAO_QRY_SUBTYPE	DSQASUBI	01	조회 하위 유형. 값은 다음과 같습니다. 1 SQL 하위 유형의 경우 2 QBE 하위 유형의 경우 3 PROMPTED 하위 유형의 경우 공백은 현재 패널이 QUERY가 아님을 의미합니다.
DSQAO_QUERY_MODEL	DSQAMODL	01	현재 조회의 모델. 값은 RELATIONAL의 경우 '1'입니다.
DSQAO_SAME_CMD	DSQACMDM	01	값은 다음과 같습니다. 0 두 개의 명령이 같지 않은 경우 1 두 개의 명령이 같은 경우

QMF 글로벌 변수

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAO_SUBSYS_ID	없음	04	<p>QMF가 TSO에서 실행중일 경우 이 값은 QMF가 접속된 지역 DB2 서브시스템의 ID입니다.</p> <p>CMS 또는 CICS로부터 DSQSUBS 프로그램 매개변수에 대한 값을 지정할 경우 이 글로벌 변수가 해당 값을 포함합니다. 이것은 매개변수가 허용되었기 때문에 발생하며 값은 처리되지 않습니다. 즉, 값은 글로벌 변수 필드에 배치되며 그 값으로 아무런 작업도 수행되지 않습니다. 이 논리는 복수 환경에서 동일한 EXEC가 사용되도록 합니다.</p>
DSQAO_SYSTEM_ID	DSQASYST	01	<p>현재 운영 체제. 값은 다음과 같습니다.</p> <ol style="list-style-type: none"> 1 VM/SP의 경우 2 MVS/SP의 경우 3 MVS/XA 또는 MVS/ESA의 경우 4 VM/XA 또는 VM/EA의 경우 5 CICS의 경우
DSQAO_TERMINATE	DSQCSESC	01	<p>QMF 종료 플래그. 값은 다음과 같습니다.</p> <ol style="list-style-type: none"> 0 세션이 표시되지 않는 경우 1 세션이 표시되는 경우
DSQAO_VARIATION	DSQAVARN	02	<p>서식 패널 변이 번호. 공백은 FORM.DETAIL이 현재 패널이 아님을 의미합니다.</p>

CICS와 연관된 DSQ 글로벌 변수

이 표의 변수 중 DSQAP_CICS_PQNAME 및 DSQAP_CICS_PQTYPE만이 SET GLOBAL 명령으로 수정될 수 있습니다.

대기열 유형이 TD이면, 해당 대기열 이름의 최대 길이는 4입니다. 예를 들어, DSQAO_CICS_SQTYPE이 TD이면, DSQAO_CICS_SQNAME의 최대 길이는 4입니다.

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAP_CICS_PQNAME	없음	08	QMF 인쇄를 포함할 CICS 데이터 대기열을 명명합니다.
DSQAP_CICS_PQTYPE	없음	02	QMF 출력을 포함하는 CICS 기억영역의 유형 TS 『보조』 기억영역에 있는 CICS 임시 기억영역 대기열에 QMF 출력을 작성합니다. 이것은 기본값입니다. TD CICS 임시 데이터 대기열에 QMF 출력을 작성합니다.
DSQAO_CICS_SQNAME	없음	08	스필 파일로 사용될 CICS 데이터 대기열을 명명합니다.
DSQAO_CICS_SQTYPE	없음	02	QMF 스플 파일을 포함하는 CICS 기억영역의 유형 TS 『보조』 기억영역에 있는 CICS 임시 기억영역 대기열에 QMF 스플 파일을 작성합니다. 이것은 기본값입니다. TD CICS 임시 데이터 대기열에 QMF 스플 파일을 작성합니다.
DSQAO_CICS_TQNAME	없음	08	QMF 추적을 포함한 CICS 데이터 대기열을 명명합니다.

QMF 글로벌 변수

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQAO_CICS_TQTYPE	없음	02	<p>QMF 추적을 포함하는 CICS 기억영역의 유형</p> <p>TS 『보조』 기억영역에 있는 CICS 임시 기억영역 대기열에 QMF 추적을 작성합니다.</p> <p>TD CICS 임시 데이터 대기열에 QMF 추적을 작성합니다. 이것은 기본값입니다.</p>

이전 명령에 의해 생성된 메시지와 관련된 DSQ 글로벌 변수

다음 글로벌 변수는 SET GLOBAL 명령에 의해 변경될 수 없습니다.

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQCM_MESSAGE	DSQCIMSG	80	메시지 텍스트
DSQCM_MSG_HELP	DSQCIMID	08	메시지 도움말 패널의 ID
DSQCM_MSG_NUMBER	DSQCIMNO	08	메시지 번호
DSQCM_SUB_TXT_ <i>nn</i>	DSQCIM <i>nn</i>	20	대체 값 <i>nn</i>
DSQCM_SUBST_VARS	DSQCIM00	04	메시지에서 대체 변수의 번호

표 편집기와 연관된 DSQ 글로벌 변수

이러한 모든 글로벌 변수는 SET GLOBAL 명령에 의해 변경될 수 있습니다.

EDIT TABLE 명령의 CONFIRM 옵션이 NO라면 표 편집기는 모든 확정 패널이 표시되지 않도록 합니다. CONFIRM 옵션이 YES이면, 표 편집기는 이 표에 표시되는 글로벌 변수 값을 검사하여 어떤 확정 범주를 활성화할 것인지 결정합니다.

표 편집기의 기본값은 EDIT TABLE 명령의 SAVE 키워드에 따라 달라집니다.

- SAVE=IMMEDIATE인 경우, 각각의 범주에 대한 기본값은 작동 가능 상태입니다.

- SAVE=END인 경우, DELETE, MODIFY 및 END/CANCEL 범주의 기본값은 활성화 상태입니다. ADD 및 CHANGE 범주의 기본값은 작동 불가능 상태입니다.

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQCP_TEADD	없음	01	ADD 하위 명령 후에 확정 패널을 표시합니다. 값은 다음과 같습니다. 0 패널이 작동 불가능합니다. 1 패널이 작동 가능합니다. 2 패널이 작동 가능 또는 작동 불가능은 표 편집기 기본값에 따라 달라집니다. 이것은 기본값입니다.
DSQCP_TECHG	없음	01	CHANGE 하위 명령 후에 확정 패널을 표시합니다. 값은 다음과 같습니다. 0 패널이 작동 불가능합니다. 1 패널이 작동 가능합니다. 2 패널이 작동 가능 또는 작동 불가능은 표 편집기 기본값에 따라 달라집니다. 이것은 기본값입니다.

QMF 글로벌 변수

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQCP_TEEND	없음	01	<p>사용자가 표 편집기 서브세션을 종료하기 위해 END 하위 명령 또는 CANCEL 하위 명령을 발행하면 확정 패널을 표시합니다. 패널은 END 또는 CANCEL이 발행되었는지 여부, 데이터베이스가 수정되었는지 여부, END 또는 CANCEL이 발행되었을 때 화면에 수정된 데이터가 포함되는지 여부에 따라 여러 가지 방식으로 나타날 수 있습니다. 값은 다음과 같습니다.</p> <p>0 패널이 작동 불가능합니다.</p> <p>1 패널이 작동 가능합니다.</p> <p>2 패널이 작동 가능 또는 작동 불가능은 표 편집기 기본값에 따라 달라집니다. 이것은 기본값입니다.</p>
DSQCP_TEDEL	없음	01	<p>DELETE 하위 명령 후에 확정 패널을 표시합니다. 값은 다음과 같습니다.</p> <p>0 패널이 작동 불가능합니다.</p> <p>1 패널이 작동 가능합니다.</p> <p>2 패널이 작동 가능 또는 작동 불가능은 표 편집기 기본값에 따라 달라집니다. 이것은 기본값입니다.</p>
DSQCP_TEDFLT	없음	01	<p>표 편집기에서 열의 기본값을 나타내는 데 사용되는 예약 문자. 초기에는 더하기 부호(+)로 설정하십시오.</p>
DSQCP_TEDFLT_DBCS	없음	04	<p>표 편집기에서 그래픽 문자열 열의 기본값을 나타내는 데 사용되는 예약 DBCS 문자. 값은 4바이트의 혼합 문자여야 하며 하나의 DBCS 문자로 구성되며, SO문자가 앞에 오고 SI 문자가 다음에 와야 합니다. 초기에는 더하기 부호(+)로 설정하십시오. 글로벌 변수가 DBCS 환경에서만 사용된다는 점에 유의하십시오.</p>

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQCP_TEMOD	없음	01	<p>표시된 데이터가 수정되고 PREVIOUS, CLEAR, SHOW CHANGE, SHOW SEARCH, REFRESH 또는 NEXT 부속 명령이 실행되면 확정 패널을 표시합니다. 결과 패널에는 패널 텍스트의 일부로 부속 명령의 이름이 포함됩니다. 값은 다음과 같습니다.</p> <p>0 패널이 작동 불가능합니다.</p> <p>1 패널이 작동 가능합니다.</p> <p>2 패널이 작동 가능 또는 작동 불가능은 표 편집기 기본값에 따라 달라집니다.</p>
DSQCP_TENULL	없음	01	<p>표 편집기에서 열의 널 값을 나타내는 데 사용되는 예약 문자. 초기에는 하이픈(-)으로 설정하십시오.</p>
DSQCP_TENULL_DBCS	없음	04	<p>표 편집기에서 그래픽 문자열 열의 널 값을 나타내는 데(또는 검색 기준 문맥에서 무시를 나타내는 데) 사용되는 예약 DBCS 문자. 값은 4바이트의 혼합 문자여야 하며, 하나의 DBCS 문자로 구성되며, SO 문자가 앞에 오고 SI 문자가 다음에 와야 합니다. 초기에는 DBCS 하이픈(-)으로 설정하십시오. 글로벌 변수가 DBCS 환경에서만 사용된다는 점에 유의하십시오.</p>

화면에 정보가 표시되는 방법을 제어하는 DSQ 글로벌 변수

이러한 모든 글로벌 변수는 SET GLOBAL 명령에 의해 변경될 수 있습니다.

QMF 글로벌 변수

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQDC_COST_EST	없음	01	<p>선택적으로 데이터베이스 비용 추정값을 억제합니다. 값은 다음과 같습니다.</p> <p>0 = no-비용 추정값을 표시하지 않습니다.</p> <p>1 = yes-비용 추정값을 표시합니다. 이것은 기본값입니다.</p>
DSQDC_CURRENCY	없음	18	<p>DC 편집 코드가 지정될 때 사용되는 통화 기호. 값은 1 - 18바이트 길이의 문자열일 수 있습니다. 영어의 경우, 기본값은 유럽 통화 기호입니다. 기본값은 기타 언어에 대해 달라집니다. DBCS 환경에서, 이 값은 SBCS 및 DBCS 문자의 혼합 문자열일 수 있습니다. SO 및 SI 문자를 포함한 혼합 문자열의 총 길이는 18바이트를 초과할 수 없습니다.</p>

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQDC_DISPLAY_RPT	DSQADPAN	01	<p>RUN QUERY 후에 보고서를 표시합니다. 값은 다음과 같습니다.</p> <p>0 QMF가 RUN 조회 명령을 통한 결과 보고서를 표시하지 않게 하려면 QMF가 DSQQMFE로 또는 BATCH 모드에서 대화식으로 시작되었을 경우 이것이 기본값입니다. BATCH 모드에서 QMF가 시작되었을 때 이 값을 변경하면 QMF 화면이 표시되지 않게 합니다.</p> <p>1 QMF가 자동으로 보고서를 표시하도록 할 경우, QMF가 호출 인터페이스로 시작되면 이것이 기본값입니다. 이것은 START 명령에서 DSQADPAN 프로그램 매개변수로 덮어쓸 수 있습니다.</p> <p>이 글로벌 변수는 응용프로그램에만 해당됩니다. RUN QUERY 명령이 명령행에 입력되면 실행되지 않습니다.</p>

QMF 글로벌 변수

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQDC_LIST_ORDER	없음	02	<p>데이터베이스 오브젝트의 목록에 오브젝트의 기본 정렬 목록을 설정합니다. 첫 번째 문자의 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> 1 목록은 기본 순서를 사용합니다. 2 목록은 오브젝트 소유자별로 정렬됩니다. 3 목록은 오브젝트 이름별로 정렬됩니다. 4 목록은 오브젝트 유형별로 정렬됩니다. 5 목록은 수정일별로 정렬됩니다. 6 목록은 마지막 사용일별로 정렬됩니다. <p>두 번째 문자의 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> A 목록은 오름차순으로 정렬됩니다. D 목록은 내림차순으로 정렬됩니다. <p>이 변수는 LIST 명령의 결과로 나열된 오브젝트에만 적용됩니다. 그것은 표시 프롬프트 패널에서와 같이 다른 문맥에서 제시된 목록에는 적용되지 않습니다. 그리고 표시 목록에도 적용되지 않습니다.</p>

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQDC_SCROLL_AMT	없음	04	<p>QMF 패널의 화면이동 크기를 설정합니다. 값은 다음과 같습니다.</p> <p>Csr 화면이동 크기를 커서로 설정합니다. 사용자가 이전, 다음, 왼쪽 또는 오른쪽으로 화면이동하는지 여부에 따라, QMF에서는 커서를 화면이동 가능 영역 맨 아래, 맨 위, 맨 왼쪽 또는 맨 오른쪽에 놓는 행 또는 열을 화면이동 시킵니다.</p> <p>Half 화면이동 가능 영역의 반으로 화면이동 크기를 설정합니다.</p> <p>Page 전체 페이지로 화면이동 크기를 설정합니다. 이것은 기본값입니다.</p> <p>n 행 또는 열 수 n으로 화면이동 크기를 설정합니다. n은 1 - 9999 사이의 수가 될 수 있습니다.</p>
DSQDC_SHOW_PANID	DSQCPDSP	01	<p>CUA와 같은 패널에 있는 표시 패널 ID. 값은 다음과 같습니다.</p> <p>0 패널 ID를 표시하지 않음. 이것은 기본값입니다.</p> <p>1 패널 ID 표시</p>

명령 및 프로시저가 실행되는 방법을 제어하는 DSQ 글로벌 변수

이러한 모든 글로벌 변수는 SET GLOBAL 명령에 의해 변경될 수 있습니다.

QMF 글로벌 변수

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQEC_ALIASES	없음	31	사용자가 MVS/ESA용 DB2 위치에서 표 목록을 요청할 때 또는 현재 서버가 MVS/ESA용 DB2 또는 워크스테이션 데이터베이스 서버인 경우 표 및 보기 별명의 목록 검색을 위한 보기
DSQEC_COLS_LDB2	없음	31	해당 위치가 DB2이면 현재 위치에 있는 표에 대한 열 정보를 검색하는 보기
DSQEC_COLS_RDB2	없음	31	원격 DB2 위치에 있는 표의 열 정보 검색용 보기(현재 위치가 아닌 경우)
DSQEC_COLS_SQL	없음	31	VM/ESA 또는 VSE/ESA용 DB2 데이터베이스에서 표의 열 정보 검색을 위한 보기.
DSQEC_FORM_LANG	없음	01	<p>저장되거나 내보낸 서식으로 기본 NLF 언어를 설정합니다. 값은 다음과 같습니다.</p> <p>0 서식은 기본 NLF 언어를 사용합니다.</p> <p>1 서식은 영어를 사용합니다. 이것은 기본값입니다.</p>

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQEC_ISOLATION	없음	01	<p>기본 조회 분리 레벨. 값은 다음과 같습니다.</p> <p>0 분리 레벨 UR, 허용안된 읽기</p> <p>1 분리 레벨 CS, 커서 안정성. 이것은 기본값입니다.</p> <p>경고: '0'으로 값을 설정하면 존재하지 않는 데이터가 QMF 보고서에 유입됩니다. QMF 보고서에 존재하지 않는 데이터가 없어야 할 경우, 값을 '0'으로 설정하지 마십시오.</p> <p>제한된 지원: QMF 7.1의 경우 값 '0'의 사용은 다음 데이터베이스 서버(SQL WITH 절을 지원하는 서버)에서만 유효합니다.</p> <ul style="list-style-type: none"> • MVS용 DB2 V4 이상 • VM/VSE용 DB2 V4 이상
DSQEC_NLFCMD_LANG	없음	01	<p>명령에 대해 예상되는 NLF 언어를 설정합니다. 값은 다음과 같습니다.</p> <p>0 명령은 기본 NLF 언어로 되어 있어야 합니다. 이것은 기본값입니다.</p> <p>1 명령은 영어로만 가능합니다.</p>

QMF 글로벌 변수

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQEC_RERUN_IPROC	없음	01	<p>END 명령 후에 호출 프로시저를 재실행합니다. 값은 다음과 같습니다.</p> <p>0 END 명령 후에 호출 프로시저의 재실행을 억제합니다.</p> <p>1 END 명령 후에 호출 프로시저를 재실행합니다. 이것은 기본값입니다.</p> <p>호출 프로시저로 QMF를 시작할 경우, 이 변수를 '0'에 설정합니다. 프로시저를 재실행하는 대신에 QMF를 종료합니다.</p>
DSQEC_RESET_RPT	없음	31	<p>임시 기억영역에 있는 불완전한 DATA 오브젝트가 성능에 영향을 미칠 때 QMF가 사용자에게 프롬프트를 표시할지 판별합니다. 가능한 값은 다음과 같습니다.</p> <p>0 보고서 프롬프트 재설정 패널은 표시되지 않으며, QMF는 실행 보고서를 완료합니다. 이것이 기본값입니다.</p> <p>1 보고서 프롬프트 재설정 패널이 표시됩니다. 이 패널은 사용자에게 완료 프롬프트를 제시하거나 새 명령어를 시작하기 전에 현재 실행중인 보고서를 재설정합니다.</p> <p>2 보고서 프롬프트 재설정 패널은 표시되지 않으며 QMF는 현재 실행중인 보고서를 재설정합니다.</p>

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQEC_SHARE	없음	31	SHARE 매개변수에 대한 기본값을 지정합니다. 가능한 값은 다음과 같습니다. 0 다른 사용자와 데이터를 공유하지 않습니다. 1 다른 사용자와 데이터를 공유합니다.
DSQEC_TABS_LDB2	없음	31	MVS/ESA용 DB2 또는 워크스테이션 데이터베이스 서버인 경우 현재 서버에서 표 및 보기 목록 검색을 위한 보기
DSQEC_TABS_RDB2	없음	31	표 목록 검색용 보기 및 원격 DB2 서브시스템의 보기
DSQEC_TABS_SQL	없음	31	VM/ESA 또는 VSE/ESA용 DB2 데이터베이스의 표 및 보기 목록 검색을 위한 보기

CONVERT QUERY의 결과를 보여주는 DSQ 글로벌 변수

다음 글로벌 변수는 SET GLOBAL 명령에 의해 변경될 수 없습니다.

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQQC_LENGTH_ <i>nnn</i>	DSQCL <i>nnn</i>	05	변환된 결과의 길이 <i>nnn</i>
DSQQC_QRY_COUNT	DSQCQCNT	03	변환된 결과의 조회의 수. 원래의 조회가 QBE I. 또는 U. 조회가 아니면 값은 항상 '1'이 됩니다.
DSQQC_QRY_LANG	DSQCQLNG	01	변환된 조회 언어. 값은 다음과 같습니다. 1 SQL의 경우 2 QBE의 경우 3 프롬프트 경우
DSQQC_QRY_TYPE	DSQCQTYP	지정되지 않음	변환 결과의 첫 단어
DSQQC_RESULT_ <i>nnn</i>	DSQCQ <i>nnn</i>	지정되지 않음	변환 결과 <i>nnn</i>

RUN QUERY 오류 메시지 정보를 보여주는 DSQ 글로벌 변수

다음 글로벌 변수는 SET GLOBAL 명령에 의해 변경될 수 없습니다.

호출 인터페이스 변수 이름	명령 인터페이스 변수 이름	길이	설명
DSQQM_MESSAGE	DSQCIQMG	80	조회 텍스트 메시지
DSQQM_MSG_HELP	DSQCIQID	08	메시지 도움말 패널의 ID
DSQQM_MSG_NUMBER	DSQCIQNO	08	메시지 번호
DSQQM_SQL_RC	DSQCISQL	16	마지막 명령 또는 조회로부터의 SQLCODE
DSQQM_SQL_STATE	없음	05	SQLSTATE가 데이터베이스 관리자에서 리턴되는 경우, DSQQM_SQL_RC에서 SQLCODE와 연관된 SQLSTATE
DSQQM_SUB_TXT_ <i>nn</i>	DSQCIQ <i>nn</i>	20	대체 값 <i>nn</i>
DSQQM_SUBST_VARS	DSQCIQ00	04	대체 변수의 번호

부록G. 주의사항

이 책은 미국에서 제공된 제품 및 서비스를 위해 개발되었습니다. IBM은 다른 나라에서는 이 책에서 논의된 제품, 서비스 또는 피처를 제공하지 않을 수 있습니다. 현재 해당 지역에서 사용이 가능한 제품 및 서비스에 대해서는 해당 지역의 IBM 영업대표에게 문의하십시오. 특정 IBM 제품, 프로그램 또는 서비스를 참조했다고 해서 이 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 의미는 아닙니다. IBM의 지적 재산을 침해하지 않는 한 동등한 어떠한 제품, 프로그램 또는 서비스도 대신 사용할 수 있습니다. 그러나, 비 IBM 제품, 프로그램 또는 서비스 운영상에 대한 평가 및 검증은 사용자의 몫입니다.

IBM은 이 책에서 논의되는 주제에 대해 특허를 갖고 있거나 현재 출원중일 수 있습니다. 이 책을 제공하는 것이 이 특허에 대한 사용권을 부여하는 것은 아닙니다. 사용권에 관해서는 서면을 통해 아래 주소로 문의하실 수 있습니다.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
지적 재산권부

2 바이트(DBCS) 정보에 관한 특허 사용권에 관한 문의는 사용자 국가의 IBM 지적 재산권부나 다음 주소로 문의하십시오.

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

다음 내용은 영국이나 이러한 조항이 지역 법규에 일치하지 않는 국가에는 적용되지 않습니다. IBM은 이 서적을 어떠한 종류의 명시적 또는 암시적 보증없이 『현상 상태 대로』 제공합니다. 여기에는 특정 목적에 대한 적합성 또는 판매 가능성, 비침해 등에 대한 암시적 보증이 포함되나, 이에 국한되는 것은 아닙니다. 일부 국

가에서는 특정 거래에 있어서 명시적 또는 암시적 보증의 거부나 허용되지 않을 수도 있습니다. 따라서, 이 내용이 사용자에게 적용되지 않을 수도 있습니다.

이 책에는 기술상의 부정확성이나 인쇄상의 오류가 있을 수 있습니다. 이 책의 내용은 정기적으로 변경되며, 이들 변경사항은 수정판에 통합됩니다. IBM은 사전 통지없이 언제든지 이 책에서 설명하는 제품 및 프로그램을 변경하거나 개선할 수 있습니다.

이 책에서 언급하는 비 IBM에 관한 참조 내용은 편의상 제공된 것이며, 어떤 방식으로든 이들 웹 사이트에 대해 보증하는 것은 아닙니다. 이들 웹 사이트에 나오는 정보는 이 IBM 제품에 대한 정보가 아니며, 이들 웹 사이트를 사용할 때 발생하는 문제는 사용자의 책임입니다.

(i) 독립적으로 작성된 프로그램(이 프로그램을 포함하여)과 기타 이 프로그램간의 정보 교환 (ii) 및 교환된 정보의 상호 사용 등의 목적으로 이 프로그램에 대한 정보를 필요로 하는 사용권자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
소프트웨어 사업부

그러한 정보는 일부 경우 사용료를 지불하여 적절한 조항 및 조건에 따라 정보를 사용할 수 있습니다.

이 책에 기술된 사용권 프로그램과 이 프로그램에 사용 가능한 모든 사용권 자료는 IBM 고객 계약, IBM 국제 프로그램 사용권 계약 또는 이와 동등한 모든 계약 조건하에 IBM에서 제공됩니다.

여기에 포함된 성능 데이터는 제어된 환경에서 결정된 것입니다. 따라서 다른 운영 환경에서는 매우 다른 결과를 나타낼 수도 있습니다. 일부 측정치는 개발 레벨의 시스템에서 작성되었을 수도 있으므로, 이들 측정치가 일반적으로 사용할 수 있는 시스템에서 동일하게 나타날 것이라는 보장은 없습니다. 더우기 일부 측정치는 추정을 통해 평가되었을 수도 있습니다. 실제 결과는 다를 수도 있습니다. 이 책의 사용자는 데이터가 해당 특정 환경에 적용 가능한 지를 확인해야 합니다.

비 IBM 제품에 관련된 정보는 해당 제품의 공급자 또는 출판물, 기타 사용할 수 있는 소스로부터 제공되었습니다. IBM은 이들 제품을 테스트하지 않았으며, 비 IBM 제품과 관련된 성능, 호환성 또는 다른 의견의 정확도를 보장할 수 없습니다. IBM 이외의 제품 기능에 대한 문의는 해당 제품 공급자에게 문의해야 합니다.

IBM의 앞으로의 방향 또는 의도에 관한 모든 내용은 통지없이 변경되거나 철회될 수 있으며, 이는 목표 및 목적만을 나타냅니다.

제시된 모든 IBM의 가격은 IBM이 제안한 소매 가격이고, 현재 시가이며, 통지없이 변경될 수 있습니다. 판매상의 가격은 이와 다를 수 있습니다.

이 정보는 계획 수립을 위한 것입니다. 이 책의 정보는 여기에서 언급된 제품이 출하되기 전에 변경될 수 있습니다.

이 책에는 일상적 업무 처리에 사용되는 데이터 및 보고서의 예가 들어 있습니다. 예를 좀더 자세히 설명하기 위해서 예제에 각 회사명, 브랜드명 및 제품명이 언급되는 경우도 있습니다. 여기에 언급된 이름은 가상의 것이므로, 실제 회사의 이름과 비슷하거나 동일한 경우에는 모두 우연의 일치입니다.

저작권:

이 정보에는 소스 언어로 된 응용프로그램이 들어 있으며 다양한 운영 플랫폼의 프로그래밍 기술을 설명하고 있습니다. 사용자는 IBM에 요금을 지불하지 않고, 예제 프로그램이 작성된 운영 체제 플랫폼의 응용프로그램 인터페이스에 부합되는 응용프로그램을 개발, 사용, 판매 또는 배포하기 위한 목적으로 어떤 형태로든 복사, 수정 및 배포할 수 있습니다. 이들 예제는 모든 조건에서 완전하게 테스트된 것은 아닙니다. 따라서 IBM은 이들 프로그램의 신뢰성, 서비스 가능성 또는 기능에 대해 보장할 수 없습니다.

이 책을 소프트카피로 보는 경우, 그림과 색상이 나타나지 않을 수도 있습니다.

등록상표

다음은 미국이나 기타 국가에 있는 IBM사의 등록상표입니다.

ACF/VTAM	IBMLink
Advanced Peer-to-Peer Networking	IMS
AIX	Language Environment
AIX/6000	MVSMVS/ESA
AS/400	MVS/XA
C/370	OfficeVision/VM
CICS	OS/2
CICS/ESA	OS/390
CICS/MVS	PL/I
CICS/VSE	PROFS
COBOL/370	QMFRACF
DATABASE 2	S/390
DataJoiner	SQL/DSVirtual Machine/Enterprise Systems Architecture
DB2	Visual Basic
DB2 Universal Database	VM/XA
Distributed Relational Database Architecture	VM/ESA
DRDA	VSE/ESA
DXT	VTAM
GDDM	
IBM	

Java 또는 모든 Java 기반 등록상표 및 로고와 Solaris는 Sun Microsystems의 등록상표입니다.

Lotus 및 1-2-3은 Lotus Development Corporation의 등록상표입니다.

Microsoft, Windows 및 Windows NT는 Microsoft Corporation의 등록상표입니다.

이중 별표(**)가 표시된 기타 회사, 제품 및 서비스명은 타사의 등록상표 또는 서비스 상표입니다.

용어집

이 용어집에서는 QMF 라이브러리 전반에 걸쳐 사용되는 용어를 정의합니다. 찾으려는 용어가 없으면, 이 책의 색인 또는 *IBM Dictionary of Computing*을 참조하십시오.

가

값(value). 표 내에 정렬된 행 및 열이 있는 데이터 요소

같음(like). 둘 이상의 유사한 또는 똑같은 IBM 운영 환경에 관계됩니다. 예를 들어, like 분산은 호환성이 있는 서버 속성 레벨의 OS/390용 DB2 UDB간의 분산입니다. 『같지않음』과 대조됩니다.

같지 않음(unlike). 둘 이상의 서로 다른 IBM 운영 환경을 참조합니다. 예를 들면, unlike 분산 VM 및 VSE용 DB2와 OS/390용 DB2 UDB 사이의 분산입니다. *like*과 대조됩니다.

게이트웨이(gateway). 다른 네트워크 구조의 두 개의 컴퓨터 네트워크를 연결하는 기능적인 단위. 게이트웨이는 같거나 유사한 구조를 가진 네트워크 또는 시스템을 연결하는 브리지와는 반대로 다른 구조의 네트워크 또는 시스템을 연결합니다.

결합(join). 같은 데이터 유형의 값을 갖는 일치된 열에 따라 둘 이상의 표로부터 데이터를 검색할 수 있는 관계형 연산

계산 변수(calculation variable). CALCid는 사용자 정의 계산값을 포함하는 서식의 특수 변수입니다. CALCid는 FORM.CALC 패널상에 정의되어 있습니다.

고객 정보 제어 시스템(CICS; Customer Information Control System). 원격 터미널에서 입력된 트랜잭션을 사용자 작성 응용프로그램에서 동시에 처리할 수 있는 IBM 사용권 프로그램. 여기에는 데이터베이스 빌드, 사용 및 유지 기능이 포함됩니다.

고정 열(fixed columns). 사용자가 수평으로 화면이동시킬 때 그 자리에 남아 있는 보고서의 열. 복수의 페이지, 인쇄된 보고서상에서 이들 열은 각 페이지의 왼쪽에 반복되어 있습니다.

고정 영역(fixed area). 고정 열이 들어 있는 보고서의 일부

공백열(unnamed column). 예제 표에 추가된 비어 있는 열. 대상표와 같이, 열을 결합하거나, 행을 결합하거나, 보고서에 상수 값을 포함시키기 위해 사용됩니다.

용어집

관계형 데이터베이스 관리 시스템(RDBMS; relational database management system). 관계형 데이터베이스 정의, 작성, 조작, 제어, 관리 및 사용을 위한 컴퓨터 기반 시스템

관계형 데이터베이스(relational database). 사용자가 표 집합으로 인식하는 데이터베이스

구조화 조회 언어(Structured Query Language). OS/390용 DB2 UDB와 VSE 또는 VM용 DB2간에 통신하는 데 사용되는 언어. 서술적인 구문으로 조회를 작성하는 데 사용됩니다.

규정자(qualifier). QMF 오브젝트를 참조할 때, 소유자를 식별하는 이름의 일부. TSO 데이터 세트를 참조할 때, 마침표로 나머지 이름과 분리된 이름의 모든 부분. 예를 들어, 'TCK', 'XYZ' 및 'QUERY'는 데이터 세트 이름 'TCK.XYZ.QUERY'의 규정자입니다.

그래픽 데이터 표시 관리자(Graphical Data Display Manager). 그래픽 기본요소에 해당하는 함수 루틴을 사용하여 그림을 절차상으로 정의하고 표시할 수 있는 루틴 그룹

그룹 행(grouped row). G. 또는 내장 함수에 의해 합산되는 QBE 목표 또는 예제 표 내의 데이터 행

근거리 통신망(LAN; local area network). (1) 국지 자원 공유를 위해 연결된 둘 이상의 프로세서 (2) 한정된 지리적 영역 내의 네트워크(예: 한 사무실 빌딩, 창고 또는 캠퍼스)

글로벌 변수(global variable). 일단 설정되면, 전체 QMF 세션에 사용될 수 있는 변수. 글로벌 변수는 프로시저, 조회 또는 서식에 사용될 수 있습니다. 런타임 변수와 대조됩니다.

기간(duration). 그 다음에 7개 키워드(YEARS, MONTHS, DAYS, HOURS, MINUTES, SECONDS, MICROSECONDS) 중 하나가 오는 숫자로 표현되는 시간의 양

기본 서식(default form). 조회 실행시 QMF에서 작성된 서식. 기본 서식은 저장된 서식이 조회와 함께 수행되는 경우에는 작성되지 않습니다.

기본 QMF 환경(base QMF environment). QMF 설치시 설정된 QMF의 영어 환경. 다른 언어 환경은 설치 후에 설정됩니다.

나

날짜(date). 일, 월, 년(3가지 값)을 지정합니다.

날짜/시간 기본 형식(date/time default formats). 날짜 및 시간 형식은 데이터베이스 관리자 설치 옵션에 의해 지정됩니다. EUR, ISO, JIS, USA 또는 LOC(LOCAL) 형식이 될 수 있습니다.

내장 오브젝트(stored object). 영구 기억영역 내에 저장된 오브젝트. 현재 오브젝트와 대조됩니다.

내장 함수(built-in function). 스칼라 함수 또는 열 함수에 대한 총칭 용어. 또한 『함수』도 될 수 있습니다.

널 값(**null value**). 널(*null*)을 참조하십시오.

널(**null**). 행에서 제공된 열에 값이 없는 경우 사용되는 특수 값. 널은 0과 같지 않습니다.

네트워크 제어 프로그램(**NCP; Network Control Program**). 단일 도메인, 복수 도메인 및 상호연결된 네트워크 기능을 지원하는 통신 제어기를 제공하는 IBM 사용권 프로그램

노드(**node**). SNA에서, 한 링크의 끝점 또는 한 네트워크에 있는 둘 이상의 링크에 연결점. 노드는 호스트 프로세스, 통신 제어기, 클러스터 제어기 또는 터미널에 분산될 수 있습니다. 노드는 경로지정과 기타 기능적인 능력에서 변할 수 있습니다.

논리 단위 유형 **6.2(LU 6.2: Logical Unit type 6.2)**. 분산 처리 환경에 있는 프로그램간의 일반적인 통신을 지원하는 SNA 논리 장치 유형

논리 단위(**LU; logical unit**). 일반 사용자가 다른 일반 사용자와 통신하기 위해 SNA 네트워크에 액세스하고, 일반 사용자가 시스템 서비스 제어점에 의해 제공된 함수를 액세스하는 포트

논리를 갖는 프로시저(**procedure with logic**). REXX 주석으로 시작되는 모든 QMF 프로시저. 논리를 갖는 프로시저에서 사용자는 조건부 논리를 수행할 수 있으며, 계산식을 만들고, 문자열을 빌드하며, 호스트 환경으로 명령을 되돌려 보낼 수 있습니다. 『선형 프로시저』를 참조하십시오.

다

다중 가상 기억장치(**Multiple Virtual Storage**). MVS/ESA 제품을 의미합니다.

단축 명령표(**command synonym table**). 각 행이 설치 정의의 명령을 설명하는 표. 각 사용자는 이들 표 중 하나로 지정될 수 있습니다.

대등 프로그램간 통신(**APPC; Advanced Program-to-Program Communication**). 상호연결된 시스템에서 통신하고 프로그램들의 처리를 공유할 수 있는 SNA 동기 데이터 링크 제어 LU 6.2 프로토콜의 구현

대상 제어표(**DCT; destination control table**). CICS에서, 임시 데이터 대기열 각각에 대한 정의가 들어 있는 표

대상표(**target table**). 예제 요소가 열을 결합하고, 행을 결합하거나, 보고서에 상수 값을 포함시키기 위해 사용되는 빈 표

대체 변수. (1) 값이 글로벌 변수 또는 실행 변수로 지정되는 프로시저 또는 조회 내의 변수 (2) 값이 글로벌 변수로 지정되는 서식 내의 변수

대화 패널(**dialog panel**). 프롬프트 조회 1차 패널의 일부를 오버레이하고 조회를 빌드하는 데 도움을 주는 대화를 확장시키는 패널

용어집

대화식 세션(interactive session). 사용자와 QMF가 상호작용할 수 있는 QMF 세션. QMF INTERACT 명령을 사용하여 다른 대화식 세션에서 시작될 수 있습니다.

대화식 실행(interactive execution). 명령 실행이 실제로 발생하는 동안 사용자와 QMF 사이에서 발생해야 하는 모든 대화 내의 QMF 명령의 실행

대화식으로 전환(interactive switch). 켜져 있을 때, 응용프로그램이 QMF 명령을 대화식으로 실행할 수 있는 개념적인 스위치

대화(conversation). LU 6.2 세션상의 두 개의 프로그램 사이에서 트랜잭션을 처리시 서로 통신할 수 있는 논리 연결

데이터베이스 관리 시스템(database management system). 데이터베이스를 정의, 생성, 조작, 제어, 관리 및 사용하기 위한 컴퓨터 기반 시스템. 데이터베이스 관리 시스템에는 데이터 무결성을 보호하기 위한 트랜잭션 관리 및 데이터 복구 기능도 있습니다.

데이터베이스 관리자(database administrator). 데이터베이스에 대한 내용과 액세스를 관리하는 사람

데이터베이스 관리자(database manager). 데이터베이스를 생성하고 유지하며, 데이터베이스에 대한 액세스를 필요로 하는 프로그램과 통신하는 데 사용되는 프로그램

데이터베이스 서버(database server). (1) DRDA에서, 응용프로그램 서버에서 받은 요청 목표 (2) OS/2에서, 데이터베이스 클라이언트에 대한 지역 데이터베이스에 데이터베이스 서비스를 제공하는 워크스테이션

데이터베이스(database). 다수의 사용자의 주문 데이터를 수용, 저장 및 제공을 위해 주어진 구조를 갖는 데이터의 집합. OS/390용 DB2 UDB에서, 표 공간 및 색인 공간이 들어 있는 작성된 오브젝트. VM 및 VSE용 DB2에서, 시스템이 유지하는 표, 색인 및 지원 정보(예: 제어 정보 및 데이터 복구 정보)의 집합. OS/2에서, 표, 보기(view) 및 색인과 같은 정보의 집합

데이터(DATA). 검색 조회에서 리턴되는 정보가 들어 있는 임시 기억영역의 오브젝트. 표에 포함되고 보고서에 형식화된 영숫자에 의해 표현되는 정보

도움말(HELP). 오류 메시지, QMF 패널 또는 QMF 명령 및 옵션에 관한 추가 정보

라

롤백(rollback). 한 응용프로그램 또는 사용자가 작성한 허용안된 데이터베이스 변경사항을 삭제하는 프로세스. 롤백이 발생하면, 잠금이 해제되고 변경되고 있는 자원의 상태는 최종 확약, 롤백 또는 초기의 상태로 리턴됩니다. 확약을 참조하십시오.

리터럴(literal). 프로그래밍 언어에서, 직접적으로 값을 나타내는 어휘 단위입니다. 값이 문자 자체에 제공된 문자열입니다.

마

매개변수(parameter). QMF 명령의 요소. 이 용어는 키워드 매개변수 또는 위치 매개변수를 언급하기 위해 QMF 문서에서 일반적으로 사용됩니다.

명령 동의어(command synonym). 설치 정의 명령의 명령어 또는 명령어/오브젝트 일부. 사용자가 다른 정보가 그 다음에 나와야 하는 명령에 대해 이를 입력합니다.

명령 인터페이스(command interface). QMF 명령을 실행하는 인터페이스. QMF 명령은 사용중인 QMF 세션 내에서만 실행될 수 있습니다. 호출 인터페이스와 대조됩니다.

문자열(string). 유사한 유형의 연속적인 항목 세트(예: 문자열(character string))

바

바인드(bind). DRDA에서, 응용프로그램의 SQL 문이 응용프로그램 지원 프로토콜(및 데이터베이스 지원 프로토콜) 흐름상에서 데이터베이스 관리 시스템에 알려지게 하는 프로세스. 바인드 동안, 사전처리 컴파일러 또는 사전처리 프로세서의 출력은 패키지라는 제어 구조로 변환됩니다. 또한, 참조된 데이터에 대한 액세스 경로가 선택되고 몇 개의 권한 검사가 수행됩니다.(OS/390용 DB2 UDB에서 선택적으로, 출력은 응용프로그램 플랜이 될 수 있습니다.)

변화(variation). 보고서 또는 보고서의 일부를 조건부로 형식화하는 데 사용될 수 있는 FORM.DETAIL 패널상에 지정된 데이터 형식화 정의

별명(alias). OS/390용 DB2 UDB에서, 동일한 또는 원격 OS/390용 DB2 UDB 서브시스템에서 표 또는 보기를 언급하기 위해 SQL 문에 사용될 수 있는 대체 이름. OS/2에서, 오브젝트, 데이터베이스 또는 LU와 같은 네트워크 자원을 식별하는 데 사용되는 대체 이름. QMF에서, 국지 또는 원격 OS/390용 DB2 UDB 서브시스템에 저장된 QMF 표 또는 보기에 액세스하는 데 사용되는 국지적으로 정의된 이름

병합(concatenation). 첫번째 문자열에 두 번째 문자열을 첨부하여 두 개의 문자열을 하나의 문자열로 결합

보고서(report). 조회에서 데이터를 검색하기 위해 실행되거나 DISPLAY 명령이 표 또는 보기에 입력되었을 때 생성되는 형식화된 데이터

보기(view). 하나 이상의 표에서의 데이터의 대체 표현. 해당 표 또는 뷰가 정의된 표에 포함된 열의 모두 또는 일부를 포함할 수 있습니다. (2) 조회에 대해 검색될 데이터의 범위를 정의하는 엔티티

분산 관계형 데이터베이스 구조(Distributed Relational Database Architecture). IBM과 공급업체 관계형 데이터베이스 제품에서 사용되는 분산 관계형 데이터베이스 처리를 위한 연결 프로토콜

분산 관계형 데이터베이스(distributed relational database). 모든 데이터가 관계형 모델에 따라 저장되어 있는 분산 데이터베이스

용어집

분산 데이터베이스(distributed database). 사용자에게는 지역적으로 액세스 가능한 하나의 논리적인 모습으로 표시되지만, 여러 장소에 있는 데이터베이스로 구성되어 있습니다.

분산 데이터(distributed data). 한 네트워크에서 둘 이상의 시스템에 저장되어 있고 원격 사용자 및 응용프로그램에 사용가능한 데이터

분산 작업 단위(distributed unit of work). 사용자 또는 응용프로그램이 단일 작업 단위 내에서 SQL 문을 SQL 문당 하나의 RDBMS만, 복수의 관계형 데이터베이스 관리 시스템에 제출할 수 있는 분산 관계형 데이터에 접근하는 방식 OS/390용 DB2 UDB은 QMF가 지원하는 시스템 직접 액세스라고 하는 그 V2R2에서 분산 작업 단위 지원의 제한된 서식을 소개합니다.

사

상관 이름(correlation name). SELECT 조회의 FROM절에 지정된 표 이름의 별명. 열 이름과 병합될 때, 열이 속한 표를 식별합니다.

상대 논리 단위(partner logical unit). SNA에서, 세션에 있는 원격 시스템

색인(index). 제공된 키로 신속하게 레코드에 액세스할 수 있는 표의 레코드 위치에 관한 데이터의 집합

샘플표(sample tables). QMF과 함께 제공되는 표. 샘플표의 데이터는 새 QMF 사용자의 제품에 대한 학습을 돕는데 사용됩니다.

서버(server). 네트워크상에서 워크스테이션에 공유된 서비스를 제공하는 기능적인 장치

서식(form). 보고서 또는 차트를 인쇄하거나 표시하기 위한 세부사항이 들어 있는 오브젝트. 임시 기억영역의 서식의 이름은 FORM입니다.

선형 구문(linear syntax). 프로그램 또는 프로시저의 한 명령문에 입력되었거나, QMF 명령행에 입력될 수 있는 QMF 명령 구문

선형 프로시저(linear procedure). REXX 명령으로 시작되지 않는 프로시저. 선형 프로시저는 QMF 명령, 주식, 공백 행, RUN 명령 및 대체 변수를 포함할 수 있습니다. 『논리를 갖는 프로시저』를 참조하십시오.

설치 지정 명령(installation-defined command). 설치에 의해 작성되는 명령. QMF에서는 이 명령을 고유의 명령 중 하나로 또는 명령 조합으로 처리합니다.

설치 지정 형식(installation-defined format). 설치로 정의된(또는 빌드된) LOCAL 형식으로도 지칭되는 날짜 및 시간 형식

세부 블록 텍스트(detail block text). 데이터의 특정 행과 연관된 보고서 본문 내의 텍스트

세부 표제 텍스트(detail heading text). 보고서 표제 내의 텍스트. 표제가 인쇄될 것인지의 여부는 FORM.DETAIL에 지정되어 있습니다.

세션(session). 사용자가 로그인한 시간부터 로그오프할 때까지 사용자와 QMF 사이의 모든 대화

소유자 이름(owner name). 제공된 오브젝트를 작성한 사용자의 권한 ID

스레드(thread). 응용프로그램의 연결을 설명하고, 진행을 추적하고, 자원 가능 처리 능력을 제공하고, OS/390용 DB2 UDB 자원 및 서비스에 대한 액세스 능력을 한정하는 OS/390용 DB2 UDB 구조. 대부분의 OS/390용 DB2 UDB 기능은 스레드 구조하에서 실행됩니다.

스칼라 함수(scalar function). 다른 값으로부터 하나의 값을 생성하고, 괄호 안에 있는 인수의 목록이 뒤에 오는 함수 이름의 서식으로 표현된 연산

스칼라(scalar). 열 내의 값, 또는 리터럴 또는 다른 스칼라를 포함하는 표현식의 값

시간소인(timestamp). 날짜 및 시간, 가능한 경우 마이크로초(6가지 또는 7가지 값)

시간(time). 하루의 시간을 시간과 분, 가능한 경우 초(2가지 또는 3가지 값)로 표시합니다.

시간/날짜 데이터(date/time data). DATE, TIME 또는 TIMESTAMP 데이터 유형인 표 열 내의 데이터

시스템 기록(SYSLOG; System Log). 작업 관련 정보, 연산 데이터, 이상 발생 설명, 명령 및 조작용 사이의 메시지가 저장될 수 있는 데이터 세트 또는 파일

시스템 네트워크 구조(Systems Network Architecture). 논리적인 구조, 형식, 프로토콜 및 정보 단위를 전송하기 위한 조작상의 순서 및 네트워크 구성 및 운영 제어에 대한 설명

실행 변수(run-time variable). 프로시저나 조회 실행시 값을 사용자가 지정한 프로시저 또는 조회 내의 변수. 실행 변수의 값은 현재의 프로시저 또는 조회에서만 사용할 수 있습니다. 글로벌 변수와 대조됩니다.

아

연결성(connectivity). 다른 시스템이 서로 통신할 수 있게 함. 예를 들면, OS/390용 DB2 UDB 응용프로그램 리퀘스터와 VM 및 VSE용 DB2 응용프로그램 서버 사이의 연결성으로 OS/390용 DB2 UDB 사용자는 VM 및 VSE용 DB2 데이터베이스에서 데이터를 요청할 수 있습니다.

열 자동 줄바꿈 기능(column wrapping). 보고서의 값이 한 열 내에 여러 줄을 차지하도록 형식화합니다. 열에 길이 가 열 너비를 초과하는 값이 들어 있는 경우에 가끔 사용됩니다.

열 함수(column function). 한 열의 모든 값에 한 번 적용되고, 결과로서 단일 값을 리턴하며, 괄호로 묶인 하나 이상의 인수가 다음에 오는 함수 이름의 서식으로 표현되는 연산

용어집

열의 레이블(column label). 데이터베이스에 저장된 데이터의 열에 대한 대체 설명자. 사용시, 열 레이블이 서식상에 기본값으로 표시되지만, 이를 사용자가 변경할 수 있습니다.

열의 표제(column heading). 사용자가 서식에 지정할 수 있는 열 이름에 대한 대안. 열 이름 및 레이블과 같이 데이터베이스에 저장되지 않습니다.

열(column). 표 데이터의 수직 세트. 특정 데이터 유형(예: 문자 또는 숫자) 및 이름이 있습니다. 열 값은 모두 같은 데이터 특성을 갖습니다.

영구 기억영역(permanent storage). 모든 표 및 QMF 오브젝트가 저장된 데이터베이스

영역 구분(area separator). 나머지 보고서 및 표시된 보고서의 고정 영역을 분리하는 관문

예제 요소(example element). QBE 조회에서 계산 및 조건에 사용되는 값의 기호

예제표(example table). QBE 조회의 기초 구조

오브젝트 이름(object name). QMF 사용자가 소유하는 오브젝트를 식별하는 문자열. 문자열은 최대 18바이트까지 가능하며, 영문자로 시작되어야 합니다. 『오브젝트 이름』이라는 용어는 『소유자 이름』 접두어를 포함하지 않습니다. 사용자는 권한이 있는 경우에만 다른 사용자의 오브젝트에 액세스할 수 있습니다.

오브젝트 패널(object panel). 한 QMF 명령의 실행 후와 다른 명령의 실행 전에 온라인으로 나타날 수 있는 QMF 패널. 이러한 패널에는 시작, 보고서, 차트 패널 및 QMF 오브젝트를 표시하는 모든 패널이 포함됩니다. 여기에는 목록, 도움말, 프롬프트 및 상태 패널이 포함되지 않습니다.

오브젝트(object). QMF 조회, 서식, 프로시저, 프로파일, 보고서, 차트, 데이터 또는 표. 보고서, 차트 및 데이터 오브젝트는 임시 기억영역에만 있으며, 데이터베이스에 저장될 수 없습니다. 표 오브젝트는 데이터베이스에만 존재합니다.

온라인 실행(online execution). 오브젝트 패널에서 또는 기능 키를 눌러 명령을 실행

워크스테이션 데이터베이스 서버(Workstation Database Server). UNIX 및 Intel 플랫폼의 IBM DRDA 데이터베이스 제품군(DB2 Universal Database(UDB), DB2 Common Server, DB2 Parallel Edition 및 DataJoiner)

원격 데이터 액세스(remote data access). 원격 위치로부터 데이터를 검색하는 방법. QMF에서 사용된 두 가지 원격 데이터 액세스 기능은 원격 작업 단위 및 시스템 직접 액세스라는 OS/390용 DB2 UDB 전용 분산 작업 단위입니다.

원격 데이터(remote data). 데이터 액세스를 시도하고 있는 서브시스템이 아닌 다른 서브시스템에 의해 유지되는 데이터. 국지 데이터와 대조됩니다.

원격 작업 단위(remote unit of work). (1) 응용프로그램이 관계형 데이터베이스와 다른 시스템에 있고 단일 논리 작업 단위 내에서 모든 원격 작업 단위가 요청하는 단일 응용프로그램 서버 서비스에 있는 SQL 분산 처리의 서식. (2) SQL 문의 원격 준비와 실행을 허용하는 작업 단위

원격(remote). 지역 관계형 DBMS와 다른 관계형 DBMS에 관계됩니다.

위치 매개변수(positional parameter). 명령 내에서 특정 위치에 배치되어야 하는 QMF 명령의 요소

위치(location). 분산 관계형 데이터베이스 시스템에 있는 특정 관계형 데이터베이스 관리 시스템. 각 OS/390용 DB2 UDB 서브시스템은 위치로 간주됩니다.

유럽 표시 형식(EUR (European) format). 다음과 같이 날짜 및 시간 값을 나타내는 형식

- 날짜: dd.mm.yyyy
- 시간: hh.mm.ss

응답 영역(echo area). 프롬프트 조치가 작성되는 프롬프트 조회 1차 패널의 일부

응용프로그램. QMF 사용권 프로그램을 수정하지 않고 QMF 기능을 확장시키는 QMF 사용자가 작성한 프로그램. QMF 프로시저를 위한 RUN 명령, 설치 정의 명령 또는 EXEC 또는 CLIST를 호출하는 CMS 또는 TSO 명령을 각각 발행 하여 QMF 세션으로부터 시작됩니다.

응용프로그램 리퀘스터(application requester). (1) 응용프로그램에서 데이터베이스 요청을 받고, 이를 응용프로그램 서버로 전달하는 기능. (2) DRDA에서, 원격 관계형 데이터베이스 관리 시스템으로의 요청 소스

응용프로그램 리퀘스터는 분산 연결의 QMF 엔드를 처리하는 DBMS 코드입니다. QMF가 접속된 국지 OS/390용 DB2 UDB 서브시스템은 OS/390용 DB2 UDB의 응용프로그램 리퀘스터가 국지 데이터베이스 관리자 내에 설치되어 있으므로 QMF용 응용프로그램 리퀘스터로 알려져 있습니다. 따라서, 전체적인 OS/390용 DB2 UDB 서브시스템(데이터를 포함하여)은 응용프로그램 리퀘스터와 연관되어 있으나, SQL 문은 현재 위치에서 처리됩니다. 이 서브시스템은 “지역 OS/390용 DB2 UDB”라고 합니다.

VM 및 VSE용 DB2를 사용하여 응용프로그램 리퀘스터는 QMF와 같은 가상 기계에서 실행됩니다. 즉, 어떠한 데이터 베이스도 VM 및 VSE용 DB2 응용프로그램 리퀘스터와 본질적으로 연관되지 않습니다.

응용프로그램 서버(application server). 응용프로그램 리퀘스터의 요청 목표. (1) 응용프로그램 프로세스가 연결된 국 지 또는 원격 데이터베이스 관리자. 응용프로그램 서버는 원하는 데이터를 포함하고 있는 시스템에서 실행됩니다. (2) DRDA에서, 응용프로그램 리퀘스터의 요청 목표. OS/390용 DB2 UDB의 경우, 응용프로그램 서버는 전체 OS/390용 DB2 UDB 서브시스템의 일부입니다.

VM 및 VSE용 DB2의 경우, 응용프로그램 서버는 VM 및 VSE용 DB2 데이터베이스 기계의 일부입니다.

응용프로그램 지원 명령(application-support command). 응용프로그램과 QMF간에 정보를 교환하기 위해 응용프로그램 내에서 사용할 수 있는 QMF 명령. 이러한 명령으로는 INTERACT, MESSAGE, STATE 및 QMF가 있습니다.

이상 종료(abend). TASK의 비정상 종료

인수(argument). 독립적인 변수

용어집

일괄처리 QMF 세션(batch QMF session). 배경으로 실행중인 QMF 세션. 지정된 QMF 프로시저가 호출될 때 시작하고, 프로시저가 끝날 때 종료됩니다. 배경 QMF 세션 동안에 사용자 대화와 패널 표시 대화는 허용되지 않습니다.

임시 기억영역 대기열(temporary storage queue). CICS에서, QMF와 응용프로그램 또는 시스템 서비스 사이의 오브젝트 전송에 사용되는 임시 기억영역

임시 기억영역(temporary storage). 현재 사용되는 조회, 서식, 프로시저, 프로파일, 보고서, 차트 및 데이터 오브젝트가 저장되어 있는 영역. 데이터 오브젝트 외에는 모두 표시될 수 있습니다.

임시 데이터 대기열(transient data queue). CICS에서, 이름이 대상 제어표(DCT)에 정의되고, 오브젝트가 후속 내부 또는 외부 처리를 위해 저장된 기억영역입니다.

자

자동 줄바꿈 기능(wrapping). 『열 자동 줄바꿈』 및 『행 자동 줄바꿈』을 참조하십시오.

자동 줄바꿈(line wrapping). 보고서의 한 행이 여러 행을 차지하도록 하는 표 행 형식화. 열 이름의 행과 열 값의 각 행은 보고서의 행 길이에 필요한 만큼 많은 행으로 나뉘어집니다.

작업 단위(unit of work). (1) 응용프로그램 프로세스 내에서 복구가능한 조작 순서. 언제든지 응용프로그램 프로세스는 하나의 작업 단위이지만, 응용프로그램 프로세스의 수명은 확약 또는 롤백 조작의 결과로서 여러 작업 단위에 관련될 수 있습니다. (2)DRDA에서, 데이터베이스 관리자가 단일 엔티티로 처리하는 SQL 명령의 순서. 데이터베이스 관리자는 작업 단위가 수행되거나 아무것도 수행되지 않을 때 데이터가 모두 변경되었는지 확인함으로써 데이터의 일관성을 확인합니다.

작업 제어(job control). VSE에서, 실행할 각 작업이나 작업 단계를 준비하도록 기억영역으로 호출된 프로그램. 기능 중 일부는 I/O 장치들 기호 이름으로 지정하고, 프로그램용으로 스위치를 설정하고, 작업 제어 명령문을 로그(또는 인쇄)하며, 각 작업 단계의 첫 단계를 폐치하기 위한 것입니다.

저장 오브젝트(saved object). 데이터베이스에 저장된 오브젝트. 현재 오브젝트와 대조됩니다.

정렬 우선순위(sort priority). 검색된 한 열의 정렬된 값으로 다른 검색된 열의 정렬 값을 판별하게 하는 검색 조회의 세부사항

조회(query). 데이터 조회 또는 조작을 수행하는 SQL 또는 QBE 문이나 프롬프트로부터 빌드된 명령문. 저장된 조회로는 SQL 조회, QBE 조회 또는 데이터베이스에 저장된 프롬프트 조회가 있습니다. 임시 기억영역의 조회의 이름은 QUERY입니다.

지역 데이터(local data). 데이터 액세스를 시도하는 서브시스템에서 관리되는 데이터. 원격 데이터와 대조됩니다.

지역(local). 사용자 프로세서에 있는 관계형 데이터베이스, 데이터 또는 파일에 관계됩니다. 『지역 OS/390용 DB2 UDB』를 참조하십시오. 원격과 대조됩니다.

지역(local) OS/390용 DB2 UDB. OS/390용 DB2 UDB와 함께, 응용프로그램 리퀘스터는 같은 MVS 시스템에서 QMF로서 실행되고 있는 OS/390용 DB2 UDB 서브시스템의 일부입니다. 따라서, 전체적인 OS/390용 DB2 UDB 서브시스템(데이터를 포함하여)은 응용프로그램 리퀘스터와 관련되어 있으나, SQL 문은 현재 위치에서 처리됩니다. 이 서브시스템은 QMF 플랜이 바인드된 장소입니다.

QMF이 TSO에서 실행될 때, 이 서브시스템은 DSQSSUBS 시동 프로그램 매개변수를 사용하여 지정됩니다. QMF가 CICS에서 실행될 때, 이 서브시스템은 자원 제어 표(RCT)에서 식별됩니다. 지역 OS/390용 DB2 UDB은 CICS 영역에서 시작된 OS/390용 DB2 UDB의 서브시스템 ID입니다.

집계 변수(aggregation variable). FORM.BREAK, FORM.CALC, FORM.DETAIL 또는 FORM.FINAL 패널을 사용하는 보고서에 있는 집계 함수. 값은 보고서가 생성시 구분 각주, 세부 블록 텍스트 또는 마지막 텍스트의 일부로서 표시됩니다.

집계 함수(aggregation function). 열 내의 데이터를 요약하는 함수들의 모든 그룹. 서식 패널에서 AVERAGE, CALC, COUNT, FIRST, LAST, MAXIMUM, MINIMUM, STDEV, SUM, CSUM, PCT, CPCT, TPCT, TCPCT와 같은 사용 코드와 함께 요청됩니다.

차

차트(chart). 보고서에서 정보의 그래픽 표시

창(window). 패널의 전부 또는 일부가 표시되는 화면의 사각형 부분. 창은 화면의 크기와 같거나 작을 수 있습니다.

초기 프로시저(initial procedure). QMF가 호출된 후 즉시 실행되는 QMF 시작 명령의 DSQSRUN 매개변수에 의해 지정된 QMF 프로시저

초기화 프로그램(initialization program). QMF 프로그램 매개변수를 설정하는 프로그램. 이 프로그램은 호출 인터페이스에서 DSQSCMD에 의해 지정됩니다. 대화식 QMF의 기본 프로그램은 DSQSCMD n 이며, 여기서 n 은 해당 언어를 위한 규정자(영어의 경우 'E')입니다.

카

클라이언트(client). 서버에서 공유 서비스를 받는 기능적인 단위

키워드 매개변수(keyword parameter). 키워드 및 할당된 값으로 구성되는 QMF 명령의 요소

타

트랜잭션 프로그램 이름(transaction program name). LU 6.2 대화에 참여하고 있는 각 프로그램이 알려진 이름. 일반적으로, 연결의 개시자는 다른 LU에서 연결하고자 하는 프로그램 이름을 식별합니다. LU 이름과 함께 사용될 때, 네트워크에 있는 특정 트랜잭션 프로그램을 식별합니다.

용어집

트랜잭션 프로그램(transaction program). SNA 네트워크에서 트랜잭션을 처리하는 프로그램. 두 종류의 트랜잭션 프로그램(응용 트랜잭션 프로그램 및 서비스 트랜잭션 프로그램)이 있습니다.

트랜잭션(transaction). '작업 단위 시작' 및 '확약' 또는 '롤백' 사이에서 발생하는 작업

과

패널(panel). 창에 표시하기 위해 함께 그룹화된 정보의 특정 정렬. 패널에는 정보용 텍스트, 입력 필드, 사용자가 선택할 수 있는 옵션 또는 이들이 혼합되어 있을 수 있습니다.

패키지(package). 응용프로그램 내의 SQL 문이 관계형 데이터베이스 관리 시스템에 바인드될 경우에 생성된 제어 구조. 데이터베이스 관리 시스템은 명령문 실행시 발견한 SQL 문을 처리하기 위해 제어 구조를 사용합니다.

표 이름 영역(table name area). QBE 예제 표의 맨 왼쪽 열

표 편집기(Table Editor). 권한이 있는 사용자가 조회를 작성하지 않고 데이터베이스를 변경할 수 있는 QMF 대화식 편집기

표의 데이터(tabular data). 열 내의 데이터. 데이터의 내용과 서식은 FORM.MAIN 및 FORM.COLUMNS에 지정되어 있습니다.

표(table). 관계형 데이터베이스 관리자의 제어하의 명명된 데이터 집합. 표는 고정된 수의 행 및 열로 구성됩니다.

프로시저. QMF 명령을 포함하는 오브젝트. 단일 RUN 명령으로 실행될 수 있습니다. 임시 기억영역 내의 프로시저의 이름은 PROC입니다. 『선형 프로시저』 및 『논리를 갖는 프로시저』를 참조하십시오.

프로시저 종료 전환(procedure termination switch). QMF MESSAGE 명령을 켤 수 있는 개념적 스위치. 켜져 있는 동안, 제어가 리턴된 모든 QMF 프로시저는 즉시 종료됩니다.

프로토콜(protocol). 통신이 수행되어야 하는 경우 반드시 따라야 하는 통신 시스템의 기능 관리 규칙

프로파일(profile). 사용자 세션의 특성에 관한 정보가 들어 있는 오브젝트. 저장된 프로파일은 영구 기억영역에 저장되었던 프로파일입니다. 임시 기억영역의 프로파일 이름은 PROFILE입니다. 각 사용자마다 하나의 프로파일만 있을 수 있습니다.

프롬프트 조회(Prompted Query). 대화 패널 세트에 대한 사용자의 응답에 따라 작성된 조회

프롬프트 패널(prompt panel). 불완전하거나 틀린 QMF 명령이 발행된 후에 표시되는 패널

플랜(plan). 여러 프로그램의 SQL 문이 플랜 작성을 위한 바인드 동안 함께 수집된 패키지의 서식

하

하위 문자열(substring). 시작 및 길이가 SUBSTR 함수에 지정된 문자열의 일부

하위 조회(subquery). 다른 조회(기본 조회 또는 상위 레벨의 하위 조회)의 WHERE 또는 HAVING 절에 나타나는 완전한 SQL 조회

함수키표(function key table). 키를 설명하는 텍스트와 함께, 하나 이상의 QMF 패널에 대한 기능 키 정의가 들어 있는 표. 각 사용자는 이들 표 중 하나로 지정될 수 있습니다.

행 연산자 영역(row operator area). QBE 목표 또는 예제 표의 가장 왼쪽 열

행(row). 표 데이터의 가로 세트

현 위치(current location). QMF 세션이 현재 연결된 응용프로그램 서버. CONNECT(응용프로그램 리퀘스터에서 처리되는)와 같은 연결 유형 명령문들을 제외하고, 이 서버는 모든 SQL 문을 처리합니다. QMF 초기화시, 현 위치는 DSQSDBNM 시동 프로그램 매개변수에 의해 표시됩니다(해당 매개변수가 지정되지 않으면, 지역 OS/390용 DB2 UDB 서브시스템).

현재 오브젝트(current object). 현재 표시된 임시 기억영역의 오브젝트. 저장 오브젝트와 대조됩니다.

호스트(host). 네트워크의 서비스를 워크스테이션에 제공하는 메인프레임 또는 중형 프로세서

호출 인터페이스(callable interface). QMF 서비스에 액세스를 제공하는 프로그래밍 인터페이스. 응용프로그램은 QMF 세션 외부에서 수행될 경우에도 이 서비스에 액세스할 수 있습니다. 명령 인터페이스와 대조됩니다.

호출 CLIST 또는 EXEC. QMF를 호출(시작)하는 프로그램

화면이동 가능 영역(scrollable area). 상, 하, 좌, 우로 이동할 수 있는 표시된 오브젝트의 보기

화면(screen). 정보가 사용자에게 표시되는 표시 장치의 물리적인 표면

확약(commit). 데이터를 영구적으로 변경하는 프로세스. 확약이 발생하면, 데이터 잠금은 다른 응용프로그램에서 확약된 데이터만 참조할 수 있도록 해제됩니다. 『구간복원』을 참조하십시오.

확장 구문(extended syntax). QMF 호출 인터페이스에서 사용된 QMF 명령 구문으로, 이 구문은 호출 인터페이스 응용프로그램에 의해 확보되고 QMF와 공유하는 기억영역에 저장된 변수를 정의합니다.

확장된 가상 기억장치(Virtual Storage Extended). 디스크 운영 체제/가상 기억장치(Disk Operating System/Virtual Storage)가 확장된 운영 체제. VSE는 (1) VSE/고급 기능 지원 및 (2) 사용자의 데이터 처리 요구를 충족시키는 데 필요한 IBM 제공 및 사용자 작성 프로그램으로 구성됩니다. VSE 및 VSE에서 제어하는 하드웨어는 완전한 컴퓨터 시스템을 구성합니다.

용어집

숫자

1바이트 문자(single-byte character). 내부 표현이 1바이트로 구성된 문자. 라틴 알파벳의 문자가 1바이트 문자의 예제입니다.

1차 패널(primary panel). 사용자 조회를 포함하는 기본 프롬프트 조회 패널

1차 QMF 세션(primary QMF session). QMF 외부에서 시작된 대화식 세션. 이 세션 내에서, 다른 세션은 INTERACT 명령을 사용하여 시작될 수 있습니다.

2단계 확약(two-phase commit). 사용중인 관계형 데이터베이스 관리 시스템이 일관성 있게 작업 단위를 확약 또는 롤백하도록 하기 위해 분산 작업 단위에서 사용되는 프로토콜

2바이트 문자 세트(DBCS; double-byte character set). 각 문자가 2바이트로 표시되는 문자 세트. 256 코드 포인트로 표시될 수 있고 2바이트 문자 세트를 필요로 하는 더 많은 기호를 포함하는 일본어, 중국어 및 한글과 같은 언어. 각 문자가 2바이트를 필요로 하므로, DBCS 문자의 입력, 표시 및 인쇄에는 DBCS를 지원하는 하드웨어 및 프로그램이 필요합니다. 1 바이트 문자 세트와 대조됩니다.

2바이트 문자(double-byte character). 2개의 문자 바이트를 필요로 하는 엔티티

3가지 이름(three-part name). 위치 이름, 소유자 ID 및 오브젝트 이름으로 구성된 표 또는 보기의 완전한 이름. 응용프로그램 서버(즉, OS/390용 DB2 UDB)에서 지원될 때, 3가지 이름은 지정된 위치에서 지정된 표 또는 보기를 검색하거나 갱신하기 위해 SQL 문 내에서 사용될 수 있습니다.

A

ABENDx. 이상 종료 문제점에 대한 키워드

AIX용 DB2. AIX용 DATABASE2. QMF의 관계형 데이터를 위한 데이터베이스 관리자

APAR. Authorized Program Analysis Report

APPC. Advanced Program-to-Program Communication

APPN(Advanced Peer-to-Peer Networking). 네트워크로 연결된 컴퓨터가 동적으로 통신할 수 있는 분산 네트워크 및 세션 제어 구조. APPC(Advanced Program-to-Program Communication)와 비교됩니다. 상호연결된 시스템에서 통신하고 프로그램들의 처리를 공유할 수 있는 SNA 동기 데이터 링크 제어 LU 6.2 프로토콜의 구현

C

CICS. 사용자 정보 제어 시스템

CMS. 대화식 모니터 시스템

CP. VM용 제어 프로그램

CSECT. 제어 섹션

D

DBCS. 2바이트 문자 세트

DBMS. 데이터베이스 관리 시스템

DOC. 문서 문제점에 대한 키워드

DRDA. 분산 관계형 데이터베이스 구조

E

EBCDIC. 확장 2진 코드화 십진 교환 코드

G

GDDM. 그래픽 데이터 표시 관리자

H

HTML. 하이퍼텍스트 마크업 언어. 월드 와이드 웹상에 표시되는 문서용 표준화된 마크업 언어

I

ICU. 대화식 차트 유틸리티

INCORROUT. 잘못된 출력에 대한 키워드

ISO(국제 표준화 기구) 형식. 다음과 같이 날짜 및 시간 값을 나타내는 형식

- 날짜: yyyy-mm-dd
- 시간: hh.mm.ss

용어집

ISPF. 대화식 시스템 생산성 기능

IXF. 통합 교환 형식. 다양한 소프트웨어 제품간의 표의 데이터를 전송하기 위한 프로토콜

J

JCL. OS/390용 작업 제어 언어

JIS(일본 산업 표준) 형식. 다음과 같이 날짜 및 시간 값을 나타내는 형식

- 날짜: yyyy-mm-dd
- 시간: hh:mm:ss

L

LOOP. 무한 루프 문제점에 대한 키워드

LU. 논리 단위

LU 6.2. 논리 단위 유형 6.2

M

MSGx. 메시지 문제점에 대한 키워드

MVS/ESA. 다중 가상 기억장치/Enterprise System Architecture(IBM 운영 체제)

N

NCP. 네트워크 제어 프로그램

NLF. 자국어 피쳐. 사용자가 미국 영어 외에 다른 언어를 선택할 수 있는 QMF에 사용할 수 있는 여러 가지 선택적인 피쳐

NLS. 자국어 지원

O

OS/390용 DB2 UDB. OS/390용 DB2 Universal Database(IBM 관계형 데이터베이스 관리 시스템).

P

PERFM. 성능 문제점에 대한 키워드

PSW. 프로그램 상태 단어

PTF. 프로그램 임시 수정

Q

QBE(Query-By-Example). 조회를 그래픽으로 작성하는 데 사용되는 언어. 자세한 내용은 *QMF 사용법*을 참조하십시오.

QMF 관리 권한(QMF administrative authority). Q.PROFILES 제어 표의 최소, 삽입 또는 삭제 권한

QMF 관리자(QMF administrator). QMF 관리 권한이 있는 QMF 사용자

QMF 명령(QMF command). QMF 언어의 일부인 모든 명령을 참조합니다. 설치 지정 명령을 포함하지 않습니다.

QMF 세션(QMF session). 사용자가 QMF를 호출한 시간부터 EXIT 명령이 발행될 때까지 사용자와 QMF 사이의 모든 대화

R

RDBMS. 관계형 데이터베이스 관리 시스템

REXX. 재구조화된 확장 실행자

S

SBCS. 1바이트 문자 세트

SNA. 시스템 네트워크 구조

SNAP 덤프(SNAP dump). QMF에서 이상 종료시 생성한 하나 이상의 기억영역의 내용에 대한 동적 덤프

SQL. 구조화 조회 언어

SQLCA. 구조화 조회 언어 통신 영역

SSF. 소프트웨어 지원 기능. 현재 모든 APAR 및 PTF에 관한 정보의 기억영역과 검색에 허용되는 IBM 온라인 데이터베이스

용어집

T

TP. 트랜잭션 프로그램

TPN. 트랜잭션 프로그램 이름

TSO. 시간 공유 옵션

U

USA(United States of America) 형식. 다음과 같이 날짜 및 시간 값을 나타내는 형식

- 날짜: mm/dd/yyyy
- 시간: hh:mm xM

V

VM. 가상 기계(IBM 운영 체제). VM/ESA 환경에 대한 총칭 용어

VSE. 확장된 가상 기억장치(IBM 운영 체제). VM/ESA 환경에 대한 총칭 용어

W

WAIT. 무한 대기 상태 문제점에 대한 키워드

참고 문헌

다음 목록에는 특정 라이브러리의 모든 책이 들어 있지 않습니다. 이들 책의 사본을 주문하거나 특정 라이브러리에 대한 자세한 정보를 얻으려면 IBM 영업대표에게 문의하십시오.

QMF 서적의 목록은 vii 페이지의 『QMF 라이브러리』를 참조하십시오.

APPC 서적

Communicating with APPC and CPI-C: A Technical Overview
Networking with APPC: An Overview

CICS 서적

OS390용 CICS 트랜잭션 서버

CICS/OS390 User's Handbook
CICS/OS390 Application Programmer Reference
CICS/OS390 Application Programming Guide
CICS/OS390 DB2 Guide
CICS/OS390 Resource Definition (Macro)
CICS/OS390 Resource Definition (Online)
CICS/OS390 Problem Determination Guide
CICS/OS390 System Definition Guide
CICS/OS390 Intercommunication Guide
CICS/OS390 Performance Tuning Handbook

VSE용 CICS

- *CICS for VSE/ESA User's Handbook*
- *CICS for VSE/ESA Application Programmer's Reference*
- *CICS for VSE/ESA Application Programming Guide*
- *CICS for VSE/ESA Resource Definition (Macro)*

- *CICS for VSE/ESA Resource Definition (Online)*
- *CICS for VSE/ESA Problem Determination Guide*
- *CICS/OS390 System Definition Guide*
- *CICS for VSE/ESA Intercommunication Guide*
- *CICS for VSE/ESA Performance Tuning Handbook*

COBOL 서적

VS COBOL II Application Programming Guide for VSE
COBOL/VSE Language Reference
COBOL/VSE Programming Guide

DATABASE 2 서적

OS390용 DB2 UDB

DB2 UDB for OS390 Installation Guide
DB2 UDB for OS390 Administration Guide
DB2 UDB for OS390 SQL Reference
DB2 UDB for OS390 Command Reference
DB2 UDB for OS390 Application Programming and SQL Guide
DB2 UDB for OS390 Message and Codes
DB2 UDB for OS390 Utility Guide and Reference
DB2 UDB for OS390 Call Level Interface Guide and Reference
DB2 UDB for OS390 Reference for Remote DRDA Requesters and Servers

VSE & VM용 DB2

DB2 Server for VM Installation Guide
DB2 Server for VSE Installation Guide
DB2 Server for VSE & VM Database Administration
DB2 Server for VM System Administration
DB2 Server for VSE System Administration
DB2 Server for VSE & VM Operation
DB2 Server for VSE & VM SQL Reference

DB2 Server for VSE & VM Application Programming
DB2 Server for VSE & VM Interactive SQL Guide and Reference
DB2 Server for VSE & VM Database Services Utility
DB2 Server for VM Message and Codes
DB2 Server for VSE Message and Codes
DB2 Server for VSE & VM Diagnostic Guide and Reference
DB2 Server for VSE & VM Performance Tuning Handbook

AS/400용 DB2

DB2 for AS/400 SQL Reference
DB2 for AS/400 SQL Programming

Parallel Edition

DB2 Parallel Edition Administration Guide and Reference

DB2 Universal Database

DB2 Universal Database Command Reference
DB2 Universal Database SQL Reference
DB2 Universal Database Message Reference

DataJoiner

DataJoiner Application Programming and SQL Reference Supplement

DCF 서적

DCF and DLF General Information

DRDA 서적

DRDA Every Manager's Guide
DRDA Connectivity Guide

DXT 서적

DXT Guide to Dialogs

Data Extract: Planning and Administration Guide for Dialogs

Data Extract: User Guide

Learning to Use DXT

그래픽 데이터 표시 관리자(GDDM) 서적

GDDM General Information

GDDM Base Programming Reference

GDDM Base Programming Guide

GDDM Guide for Users

GDDM Installation and System Management for VSE

GDDM Messages

HLASM 서적

IBM High-Level Assembler Programmer's Guide for OS/390, VM and VSE

IBM High-Level Assembler Language Reference for OS/390, VM and VSE

ISPF/PDF 서적

OS/390

Interactive System Productivity Facility for OS/390 Installation and Customization

Interactive System Productivity Facility for OS/390 Dialog Management Guide

Interactive System Productivity Facility for OS/390 Dialog Management Services and Examples

VM

ISPF for VM Dialog Management Services and Examples

OS/390 서적

유틸리티

OS/390 Administration: Utilities

OS/390 Extended Architecture Utilities

JCL

OS/390 Extended Architecture JCL Reference

OS/390 Extended Architecture JCL User's Guide

OS/390 JCL Reference

OS/390 JCL Users Guide

PLPA(Pageable Link Pack Area)

OS/390 Extended Architecture Initialization and Tuning

OS/390 SPL: Initialization and Tuning

VSAM

OS/390 VSAM Administration Guide

OS/390 VSAM Catalog Administration Access Method Services

TSO

OS/390 TSO Primer

OS/390 User's Guide

SMP/E

OS/390 System Modification Program Extended Messages and Codes

OS/390 System Modification Program Extended Primer

OS/390 System Modification Program Extended Reference

OS/390 System Modification Program Extended User's Guide

PL/I 서적

PL/I VSE Language Reference

PL/I VSE Programming Guide

REXX 서적

OS/390 환경

IBM Compiler and Library for REXX/370: User Guide and Reference
TSO Extensions REXX/MVS Reference

VM 환경

Procedures Language VM/REXX Reference
Procedures Language VM/REXX User's Guide

ServiceLink 서적

ServiceLink User's Guide

VM 서적

Virtual Machine Planning Guide and Reference
Virtual Machine CMS Command and Macro Reference

VSE 서적

VSE Planning Guide
VSE Guide to System Functions
VSE System Utilities
VSE Guide for Solving Problems

색인

[가]

가져오기 89
 날짜/시간 정보 120
 데이터 오브젝트 95
 변환된 서식 124
 비 QMF 오브젝트 90
 서식 오브젝트 98, 120, 123
 오류 검출 96
 오류 처리 125
 오브젝트 레벨 정보 101
 차트 오브젝트 98
 코드화 형식 오브젝트 112
 표 오브젝트 95
 프로시저 97
 프롬프트 조회 오브젝트 98, 115
 QMF 외부에서 작성된 표 92
 SQL 조회 97
값(V) 레코드 104
고급 어셈블러(HLASM) 148
관련된 제품 7
구분
 패널 120, 256
구분 패널 120, 256
그래픽 데이터 형식(GDF) 98
글로벌 변수
 규칙의 경우 76
 변수 작성 73
 설정 74
 SET GLOBAL 명령 74
 엑세스 65
 작성 73
 DSQDC_DISPLAY_RPT 85
 DSQEC_RERUN_IPROC 62
 QMF는 RUW를 통해 사용됩니다
 286

[나]

날짜/시간
 정보 121
날짜/시간 정보 121
내보내기
 가로 보고서 133
 구분 패널 120
 날짜/시간 정보 121
 내용 117
 데이터 오브젝트 92
 릴리스 특정의 형식 126
 변환된 서식 127
 보고서 오브젝트 92, 128
 서식 오브젝트 89, 117
 오브젝트 유형 89
 차트 오브젝트 98
 코드화 형식 오브젝트 112
 표 오브젝트 92
 프로시저 오브젝트 97
 프롬프트 조회 오브젝트 98, 114
 형식 89
 2진 데이터 96
 EBCDIC 데이터 96
 SQL 조회 97
내보내기된 오브젝트에서 제어 영역
 보고서 파일의 레코드 103
 서식 파일의 레코드 103
내보낸 오브젝트에서 제어 영역 103
 T 레코드 106
 V 레코드 104

[다]

다국어 환경 54

대체

변수
 값 할당 12
 구문 12
 글로벌 변수, 설정 12

대체 변수 13

값 할당 12
구문 12
글로벌 변수, 설정 12
REXX 프로그램 호출 19
RUN 명령 13

대화식 모드

명령 실행 69
초기 프로시저 10
GDDM ICU 66
QMF 66

대화식 전환(DSQAQO_INTERACT) 69

대화식 차트 유틸리티(ICU) 8

데이터

값 레코드(V) 104
레코드, 내보내기 253
연속(C) 레코드 132
오브젝트
 가져오기 95
 가져오기 오류 96
 가져오기/내보내기 규칙 95
 가져오기/내보내기 파일 세부사항
 137
 내보낸 92, 93
 헤더 93
 형식 92, 251
 2진 데이터 96
 IXF 내보내기된 형식 265
유형 너비 253
전송 속도 96

데이터 (계속)
 표 설명 레코드(T) 105
 표 행 레코드(R) 109
 2진 96
 D 레코드 268

데이터 유형
 가져온 서식 내부 126
 키워드 126

데이터 추출(DXT) 8

데이터베이스 90
 비 QMF 오브젝트, 저장 90
 원격 연결 11
 프롬프트 조회 오브젝트, 가져오기 113

데이터베이스 원격 연결 11

동의어 84

다버깅에 대한 A 옵션 142
 다버깅에 대한 L 옵션 141

[라]

레코드
 가변 형식 103
 고정 형식 99
 공유 값 132
 데이터 값(V) 104
 데이터 연속(C) 132
 데이터(D) 268
 보고서 행(L) 130
 순서, 프롬프트 조회 파일 115
 열(C) 267
 오브젝트의 종료(E) 111
 응용프로그램 데이터(*) 111
 표 설명(T) 105, 266
 표 행(R) 109
 헤더 99
 형식 253

루프, 연속 가능 10

리턴 코드
 메시지 17
 명령 인터페이스 39, 40

리턴 코드 (계속)
 호출 인터페이스 25
 0이 아님 16
 REXX 명령 28

[마]

마이그레이션
 구분 필드 번호 121
 서식 오브젝트 122
 어셈블러 151
 오브젝트 레벨 101
 C 175
 COBOL 193
 PL/I 227

마이그레이션 정보 101, 121, 122, 151, 175, 193, 227

매크로, 제품 인터페이스 283

메시지
 종료 17
 추적 데이터 출력 작성 141, 144
 표시 70
 DSQCOMM 146

명령
 길이 22
 대화식 실행 69
 리턴 코드 16
 시스템 고유 12
 언어 변수 52
 원격 작업 단위 59
 응용프로그램 59
 인터페이스 4
 리턴 코드 38
 샘플 프로그램 34
 설명 33
 프로그램 33
 프로그램에서 호출하려면 35
 필요조건 5
 DSQCCI 33
 END 명령 36

명령 (계속)
 인터페이스 4 (계속)
 INTERACT 명령 69
 SELECT 서비스 35

처리 정보 22

추적 데이터 출력 작성 141

호출 인터페이스 26

호출 인터페이스를 통한 전달 22

환경 281

2개 언어 응용프로그램 54

ADDRESS 281
 CONNECT 59
 DSQCIX 서브루틴 244
 EDIT 47
 ISPF로 47
 END 61
 EXIT 64
 GET GLOBAL 65
 ICU 66
 INTERACT 66
 ISPF 46
 LAYOUT 117
 MESSAGE 70
 QMF 시작 62
 QMF 제공 인터페이스 루틴 21
 REXX 리턴 코드 28
 RUN 13
 SAA 조회 244
 SELECT 45
 SET GLOBAL 73, 74
 START 27, 77
 STATE 36

명령 동의어
 보기 4
 사용 84
 설명 7
 작성 84
 표 85
 형식 84
 IRM 8

명령 동의어 (계속)

NLF 표 56

목록화된 프로시저

C 182

CICS 제공 163

COBOL 200

DFHEBTAL 163

PL/I 233, 234

[바]

번역할 수 있는 응용프로그램 57

변수

규칙 76

글로벌 12, 285

대체 12

대체 12

데이터 22

명령 언어 52

설정 12

설정, 글로벌 74

엑세스, 글로벌 65

언어 감지 오브젝트 57

오류 처리 28

인터페이스 통신 23

조회 내부 44

폴 22, 36

프롬프트 14

형식 레코드 103

호출 인터페이스로부터 제공 19

QMF 36

rc 16

REXX 15, 241

변이 패널 123, 127

보고서

가로 261

가져오기 규칙 112

가져오기/내보내기 파일 세부사항
137

내보내기 92

보고서 (계속)

내보내기 규칙 112

내보내기 레코드 129

내보내기 보기 128

내보내기 사용 127

내보내기 형식 261

내보낸 가로 133

내용 99

데이터 형식 251

레코드 99

레코드 유형 129

샘플 헤더 102

소형 세션 85

억제 68

오브젝트 3

가로 261

가져오기 규칙 112

가져오기/내보내기 파일 세부사항
137

내보내기 규칙 112

내보내기 형식 261

내용 99

샘플 헤더 102

표 데이터 105

표 번호 261

필드 값 104

필드 번호 261

행 데이터 109

형식 98

텍스트 표시 71

패널 3

표 데이터 105

표 번호 261

표시 68

필드 값 104

필드 번호 261

행 데이터 109

행(L) 레코드 130

형식 98

HTML 92, 135, 263

불완전한 데이터 프롬프트 96

[사]

서식

가져오기

규칙 112

기본 파일 120

오류 125

코드화 형식 98

파일 세부사항 137

필드 123

기본, 작성 117

내보내기 98, 112

개요 117

규칙 112

릴리스 특정의 형식 126

코드화 형식 98

파일 세부사항 137

형식 256

내용 99

데이터 유형 키워드 126

데이터 형식 251

레벨, 내보내기 122

변이 번호 123, 127

변환 124, 127

샘플 헤더 102

설명 98

응용프로그램 마이그레이션 지원 122

응용프로그램 필요조건 120

파일 레코드 98

표 데이터 105

표 번호 256

표시 영역 117

필드 값 104

필드 번호 256

행 데이터 109

QMF 외부 120

서식 오브젝트 98, 99, 102, 104, 105,
109, 117, 120, 122, 123, 124, 126,
127, 251, 256

선형 프로시저 71
 억제 71
 STOPPROC 옵션 71
 세션 환경 55
 소형 세션
 보고서 85
 유효하지 않은 명령 87
 유효한 명령 87
 시스템 응용프로그램 구조(SAA)
 언어 지원 5
 응용프로그램 5
 조회 명령 23, 244
 프로그램 이식성 23
 호출 인터페이스 21, 147
 START 명령 키워드 78
 시작 패널 10, 62

[아]

어셈블러
 고급 어셈블러(HLASM) 148
 마이그레이션 정보 151
 매크로 283
 샘플 프로그램 152
 언어 인터페이스 148
 인터페이스 통신 영역 맵핑
 (DSQCOMMA) 148
 통신 영역 162
 함수 호출 149
 CICS 148
 샘플 프로그램 153
 MVS 163
 VSE 164
 CMS 샘플 프로그램 157, 166
 TSO 샘플 프로그램 157, 168
 어셈블러 H(Assembler H) 148
 언어
 리턴 코드 28
 변수(DSQEC_NLFCMD_LANG)
 52
 호출 인터페이스 5

언어 (계속)
 AH(Assembler H) 148
 C 171
 COBOL 189
 FORTRAN 207
 HLASM 148
 ID 43
 NLF 56
 PL/I 224
 QMF 제공 인터페이스 루틴 21
 REXX 241
 START 명령 구문 77
 열 267, 268
 데이터 형식 268
 C 레코드 267
 오류
 가져오기 96
 가져오기 서식 125
 가져오기(데이터 및 표 오브젝트) 96
 가져오기(서식) 125
 레이블 16
 메시지 17
 명령에 대한 신호 16
 불완전한 데이터 프롬프트 95
 서버루틴으로 분기 16
 연속 가능 루프 10, 62
 처리 명령문, REXX 16
 EXEC 또는 CLIST에서 발견 및 분
 석 39
 REXX 변수를 사용하는 처리 28
 START 명령 146
 오류 명령어에 대한 신호 16
 오브젝트 4
 가져오기 4
 데이터 오브젝트 95
 오류 96
 차트 98
 코드화 형식 112
 표 오브젝트 95
 형식 251

오브젝트 4 (계속)
 내보내기 4
 내용 99
 데이터 유형 96
 사용 90
 유형 89
 차트 98
 코드화 형식 112
 형식 89, 92, 251
 IXF 형식 265
 데이터 92
 레벨 101, 120
 비 QMF, 저장 90
 서식 98, 120
 외부화 형식 90
 이식 가능 90
 작성 90
 저장 90
 전송 90
 종료 111
 표 92
 프로시저 4, 97
 프롬프트 조회 98
 2개 언어 51
 SQL 조회 97
 오브젝트의 종료 레코드(E) 111
 외부화 형식 90
 원격
 데이터베이스 연결 11
 작업 단위(UOW) 11
 프로시저 11
 원격 작업 단위
 명령 동작 59
 응용프로그램 2, 3, 4, 27, 77
 개발 1
 구현 방법 4
 논리를 갖는 프로시저 4
 데이터 레코드 111, 120
 디버깅 141
 명령 1

응용프로그램 2, 3, 4, 27, 77 (계속)

- 개요 59
- 처리 22
- END 61
- EXIT 64
- INTERACT 66
- MESSAGE 70
- 명령 동의어 4
- 변환 57
- 서식 오브젝트 120
- 시작 27, 77
- 오류 처리 28
- 유형 1
- 제어 2
- 프로시저 9
- 2개 언어 51, 90
- CICS 환경 5
- ISPF 필요조건 43
- QMF
 - 구현 방법 4
 - 시작 27, 77
 - 제어 2
 - QMF에서 실행 3
- QMF에서 실행 3
- QMF와의 프로그래밍 인터페이스 5
- REXX 프로그램 호출 18
- SAA 5
- 응용프로그램 디버깅
 - 추적에 대한 A 옵션 사용 142
 - 추적에 대한 L 옵션 141
 - 파일 할당 144
 - ISPF, 사용 48
 - PDF 대화 검사 48
 - START 명령 오류 146
 - TRACE 옵션 141
- 응용프로그램 시스템(AS) 7
- 이식성 23
- 인수 16
- 인터페이스
 - 매크로 283

인터페이스 (계속)

- 명령 4
 - 리턴 코드 38
 - 샘플 프로그램 34
 - 설명 33
 - 호출 33, 35
 - END 명령 36
- 사용자 정의 43
- 통신 매크로 23
- 통신 변수 4, 23, 241
- 통신 영역 4
 - 변경 23
 - 어셈블러 148
 - 정의 23
 - 처리 정보 22
 - 필드 설정 24
 - C 172
 - COBOL 190
 - FORTRAN 207
 - PL/I 224
- 프로그래밍 283
- 호출 가능
 - 설명 5, 21
 - 함수 21
- REXX 4
- EXEC CICS LINK 151
- REXX CALL 247
- 일괄처리 모드 63
 - END 명령 63
- 임시 기억영역 4, 28
 - 대기열 139
 - 변경 28
 - 제한사항 4
- 임시 기억영역 대기열 139
- 임시 데이터 대기열 139
 - 임시 기억영역 대기열과 대조됩니다. 139

[자]

- 자국어 기능(NLF) 54
- 자원 제어 표 31
- 제품 인터페이스 매크로 283
- 조회
 - 내보내기 형식
 - 프롬프트 98
 - QBE 136
 - SQL 97
 - 편집 47, 97
 - 포함하는 변수 44
 - QMF 외부에서 작성 97
- 주석 111, 121
 - 내보내기 형식 121
 - 응용프로그램 데이터 레코드 111, 121
- 주의사항 307

[차]

- 차트
 - 오브젝트 98
 - 차트 오브젝트 98
 - 초기 프로시저 62
 - 대화식 모드 10
 - 반복 62
 - 시작 패널 62
 - 이름, 지정 10
 - 작성 10
 - 재실행 10
 - 저장 11
 - 지정 62
 - 2개 언어 응용프로그램 53
 - CONNECT 명령 11
 - DSQEC_RERUN_IPROC 글로벌 변수 10
 - END 명령 62
- 추적
 - 데이터
 - 파일 144

추적 (계속)

- 데이터 출력 142
- 보기 145
- 추적 정의 작성 48
- 추적 해제 143
- A 옵션 142
- ISPF 명령 48
- L 옵션 141
- MESSAGE 명령 144
- 추적 데이터 출력 142, 144

[카]

코드화 형식

- 가로 보고서 133
- 가져오기/내보내기 파일 세부사항 137
- 보고서 오브젝트 128
- 사용 98
- 오브젝트 89
 - 가져오기 규칙 112
 - 내보내기 규칙 112
 - 형식 251
- 정보 구성 105
- 정의 89
- 키워드 78, 126
- 데이터 유형 126
- START 명령 78

[타]

텍스트 표시 71

통신 영역

- 어셈블러 148, 162
- 정의 23
- C 172, 180
- COBOL 190, 196
- FORTTRAN 207, 215
- PL/I 224, 231

[파]

패널

- 구분 120
- 대화식 66
- 변이 123, 127
- 사용자 정의 43
- 시작 62
- 현재 66
- NLF 필요조건 56

편집 코드 126

표

- 가져오기 92
 - 규칙 95
 - 오류 96
- 파일 세부사항 137
- 명령 동의어 85
- 보고서, 번호 261
- 서식, 번호 256
- 설명 레코드(T) 105, 266
- 오브젝트
 - 가져오기 92, 95
 - 가져오기 오류 96
 - 가져오기/내보내기 규칙 95
 - 가져오기/내보내기 파일 세부사항 137
 - 정의 92
 - 처리 92
- 작성 92
- 프롬프트 조회 번호 254
- 행 레코드(R) 109
- QMF 외부에서 작성 265

표 레코드 266

프로그램 호출 18, 22

프로시저

- 가져오기 97
- 가져오기/내보내기 파일 세부사항 137
- 값 제공 16
- 내보내기 97

프로시저 (계속)

로직을 가진

- 글로벌 변수 73
- 대체 변수 19
- 변수 12
- 변수 프롬프트 14
- 사용 4
- 이점 4
- 작성 9
- 종료 17
- ISPF 사용 46
- REXX 프로그램 호출 18

명령 동의어 7

목록에 있는 4

C 182

CICS 제공 163

COBOL 200

PL/I 233, 234

복수 조회 68

선형

변수 프롬프트 14

사용 4

시스템 고유 명령 12

원격 작업 단위 11

인수 16

임시 기억영역 4

종료 스위치 71

초기 4

편집 47

CONNECT 명령 11

QMF 외부에서 작성 97

REXX 4

프롬프트 조회

가져오기 98, 115

가져오기 규칙 112

가져오기/내보내기 파일 세부사항 137

관계 113

내보내기 98, 114

내보내기 규칙 112

프롬프트 조회 (계속)

- 내보내기 형식 254
- 내용 99
- 데이터 형식 251
- 레코드 순서 115
- 샘플 헤더 101
- 설명 98
- 파일 레코드 98
- 표 데이터 105
- 표 번호 254
- 필드 값 104
- 필드 번호 254
- 행 데이터 109

프롬프트 조회 오브젝트 98, 99, 101, 104, 105, 109, 112, 113, 114, 115, 137, 251, 254

[하]

함수 호출

- 어셈블러 149
- C 173
- COBOL 191
- DSQCIA 149
- DSQCIB 191
- DSQCIC 173
- DSQCICE 173
- DSQCIF 208
- DSQCIFE 208
- DSQCIPL 225
- DSQCIPX 225
- DSQCIX 244
- DSQCIX 서브루틴 244
- FORTTRAN 208
- PL/I 225
- REXX 18, 244

향상된 연결성 기능(ECF) 8

헤더 레코드

- 길이, 계산 93
- 보고서 오브젝트 102
- 서식 오브젝트 102

헤더 레코드 (계속)

- 오브젝트 레벨 101
- 정보 99
- 프롬프트 조회 오브젝트 101
- 필드 99
- 형식 251, 252
- IXF 266
- 현재 위치 59
- 형식
 - 가져오기 251
 - 내보내기 251
 - 데이터 92
 - 데이터 오브젝트 해석 93
 - 데이터, 내보내기 253
 - 보고서 오브젝트 128, 261
 - 서식 오브젝트 256
 - 열 데이터 268
 - 오브젝트 레벨 101
 - 외부화 90
 - 코드화
 - 사용 98
 - 오브젝트 가져오기 규칙 112
 - 오브젝트 내보내기 규칙 112
 - 정보 구성 105
 - 정의 89
 - 표 92
 - 프롬프트 조회 오브젝트 254
 - 헤더 레코드 251, 252
 - IXF 89, 265
- 호출 인터페이스
 - 논리를 갖는 프로시저로부터 호출 19
 - 리턴 코드 25
 - 매크로 283
 - 명령 26
 - 명령 처리 정보 22
 - 변수 제공 19
 - 샘플 프로그램 4
 - 어셈블러 152
 - C 176
 - COBOL 194

호출 인터페이스 (계속)

- 샘플 프로그램 4 (계속)
 - FORTTRAN 210
 - PL/I 227
 - REXX 245
- 설명 5, 21
- 언어 5, 21, 147
- 응용프로그램 디버깅 141
- 응용프로그램, 실행 27
- 이점 5
- 인터페이스 호출 23
- 통신 영역 4
 - 변경 23
 - 어셈블러 148, 162
 - 오류 처리 28
 - 정의 23
 - 필드 설정 24
 - C 171, 180
 - COBOL 196
 - FORTTRAN 207, 215
 - PL/I 224, 231
- 프로그램 4
 - CICS, 실행 30
 - COBOL 189
 - FORTTRAN 207
 - GET GLOBAL 명령 65
 - ISPF 4
 - MVS에 있는 CICS 163
 - PL/I 224
 - QMF 변수 사용 65
 - REXX
 - 사용 4
 - 설명 241
 - 통신 변수 241
 - 프로그램 호출 18
 - 호출 9
 - QMF 시동 27
 - SET GLOBAL 명령 74
 - START 명령 4
 - 구문 77, 78

호출 인터페이스 (계속)
QMF 시작 27, 77
환경 5

[숫자]

2개 언어 오브젝트 51
2개 언어 지원
오브젝트 51
2진 데이터 96

A

ADDRESS 명령 18, 281
ARG 문 16
AS(응용프로그램 시스템) 7

C

C 언어
마이그레이션 정보 175
샘플 프로그램 176
인터페이스 필요조건 175
통신 영역 171
기본값 172
맵핑 172
DSQCOMM 180
함수 호출 173
호출 인터페이스 171
CICS 181
CMS 184
ISPF 186
TSO 186
CALL 명령 18
CICS
데이터 대기열 5
임시 기억영역 대기열 138
임시 데이터 대기열 138
IXF 형식 265
QMF 오브젝트 전송에 사용 91
어셈블러 5

CICS (계속)
샘플 프로그램 153
MVS 필요조건 163
VSE 필요조건 164
영역 31
임시 기억영역 대기열 138
임시 데이터 대기열 138
프로그램은 매개변수 덮어쓰기를 시작
함 30
호출 인터페이스 5
31 비트 주소 148
C 프로그램 181
COBOL 프로그램 199
CONNECT 명령 12
DB2 대화 31
EXEC CICS LINK 인터페이스
151
IMPORT나 EXPORT 사용시 고려사
항 138
PL/I 233, 234
VSE/ESA
가져오기/내보내기 파일 속성 137
어셈블러 164
C 프로그램 182
COBOL 프로그램 200
HLASM 프로그램 164
CMS
샘플 어셈블러 프로그램 157
어셈블러 프로그램 166
C 프로그램 184
COBOL 프로그램 202
FORTRAN 217
PL/I 235
REXX 프로그램 247
COBOL
구분문자 198
마이그레이션 정보 193
매크로 283
샘플 프로그램 194
실행 필요조건 198

COBOL (계속)
통신 영역 190
함수 호출 191
호출 인터페이스 189
CICS 199, 200
DSQCOMM 196
ISPF 204
TSO 204
VM 202
VSE 200
CONNECT 명령
보기 59
설명 59
초기 프로시저 11
프로시저 11
SQL/DS 12
VM 12
C(데이터 연속) 레코드 132
D
DB2(IBM DATABASE 2)
원격 연결 11
CICS 필요조건 31
CONNECT 명령 12
DSQABFA 157
DSQABFAC 153
DSQADPAN 80
DSQALANG 80
DSQCIA 149
DSQCIX 서브루틴 244
DSQCOMM
메시지 텍스트 146
어셈블러 148, 162
오류 처리 28
정의 23
필드 설정 24
C 172, 180
COBOL 190, 196
DSQCOMM 148, 162
DSQCOMMB 196

DSQCOMM (계속)

DSQCOMM 180
 DSQCOMM 207, 215
 DSQCOMM 224, 231
 FORTRAN 207, 215
 PL/I 224, 231
 DSQDC_DISPLAY_RPT 글로벌 변수
 85
 DSQEC_RERUN_IPROC 글로벌 변수
 10
 DSQRUN 62
 DSQSBSTG 81
 DSQSCMD 81
 DSQSDBCS 81
 DSQSDBNM 82
 DSQSDBQN 82
 DSQSDBQT 82
 DSQSDEBUG 82
 DSQSDCSS 82
 DSQSIROW 82
 DSQSMODE 83
 DSQSPILL 83
 DSQSPRID 83
 DSQSRSTG 83
 DSQSRUN 83
 DSQSSPQN 83
 DSQSSUBS 83
 DSQSUSER 84
 DXT(데이터 추출) 8

E

ECF(향상된 연결성 기능) 8
 EDIT 명령 47
 END 명령
 대화식 세션 249
 명령 인터페이스 36
 설명 61
 세션 유형 61
 일괄처리 모드 63
 초기 프로시저 62

END 명령 (계속)

세션 유형 61 (계속)
 초기 프로시저가 없는 경우 63
 호출 인터페이스 시동 62
 INTERACT 명령 63
 초기 프로시저 재실행 10
 EXIT 명령 64
 EXPORT 명령
 데이터 오브젝트 92
 표 오브젝트 92
 CICS 사용 138
 DATA 91
 IXF 옵션 265

F

FORTRAN

매크로 283
 샘플 프로그램 210
 통신 영역 207
 함수 호출 208
 호출 인터페이스 207
 CMS 217
 DSQABFF 210
 DSQCOMM 215
 ISPF 220
 MVS 219
 TSO 220

G

GDDM(Graphical Data Display
 Manager) 8, 66
 대화식 차트 유틸리티 8
 GET GLOBAL 명령 26, 65

H

HTML 보고서 92, 135, 263

I

ICU(Interactive Chart Utility) 8, 66

IFX

OUTPUTMODE=BINARY 278

IMPORT 명령

실행중의 오류 및 경고 125
 정의 89
 CICS 사용 138
 DATA 옵션 95

INTERACT 명령

명령 서식 69
 설명 66
 세션
 서식 66
 시동 63
 종료 63, 64, 68

ISPF(Interactive System Productivity Facility)

명령 45
 변수 풀 36
 사용자에게 의해 작성된 패널, 표시 70
 어셈블러 프로그램 169
 응용프로그램 디버깅 48
 추적 명령 48
 패널 생성 8
 호출 인터페이스 43, 45
 COBOL 프로그램 204
 EDIT 명령 47
 FORTRAN 220
 PL/I 238
 QMF 시작 62
 SELECT 명령 45
 SELECT 서비스 35
 TSO/C 프로그램 186

IXF

샘플 레코드 276
 OUTPUTMODE=CHARACTER 276

IXF(Integrated Exchange Format)

문자 269
 2진 96, 269

L

LAYOUT 명령 117
Lotus 1-2-3/M 8
L(보고서 행) 레코드 130

M

MESSAGE 명령
보기 73
선형 프로시저 실행 억제 71
설명 70
옵션 70
추적 144
텍스트 표시 71
ISPF 패널 70
QMF 도움말 패널 70
REXX EXIT 명령 17

N

NLF(National Language Feature)
다국어 환경 54
명령 동의어 표 56
세션 환경 55
언어 56
언어 ID 43
응용프로그램 번역 57
정의 51
패널 필요조건 56
프로파일 매개변수 56

P

PARSE ARG 문 16
PDF 47, 48
PL/I
마이그레이션 정보 227
매크로 283
샘플 프로그램 227
통신 영역 224

PL/I (계속)

함수 호출 225
호출 인터페이스 224
CICS 233, 234
DSQABFP 227
DSQCOMM 231
ISPF 238
MVS 232
TSO 237
VM 235

Q

QBE(Query-By-Example) 136, 137
가져오기/내보내기 파일 세부사항
137
내보내기 형식 136

R

Repository Manager/OS/390 8
REXX

논리를 갖는 프로시저
대체 변수 19
오류 처리 명령문 16
이점 4
작성 9
데이터 처리 속도 96
매크로 283
명령 리턴 코드 28
명령 환경 281
변수 12, 15
설명적인 4
인터페이스 통신 변수 23, 241
컴파일러 4
프로그램 호출 18
함수 호출 244
호출 18
호출 인터페이스 4
샘플 프로그램 245

REXX (계속)

호출 인터페이스 4 (계속)
설명 241
액세스 244
오류 처리 변수 28
CMS, 프로그램 실행 247
QMF 시동 27
QMF로부터의 호출 45
START 명령 78
TSO, 프로그램 실행 248
DSQABFX 프로그램 245
END 명령 249
EXIT 명령 17
INTERACT 루핑 249
MESSAGE 명령 17
RUN 명령
값 제공 16
대체 변수 12, 13
삼입된 대체 변수 19
프롬프트 패널 13
ARG 옵션 16
RUW(원격 작업 단위(UOW)) 11

S

SAA(시스템 응용프로그램 구조)
언어 지원 5
응용프로그램 5
조회 명령 23, 244
프로그램 이식성 23
호출 인터페이스 21, 147
START 명령 키워드 78
SAVE DATA 명령 91
SELECT 명령 33, 45
SET GLOBAL 명령
구문 74
변수 프롬프트 15
호출 인터페이스 26, 74
SQL 조회 오브젝트 97, 137

SQL(Structured Query Language)

조회

오브젝트 97, 137

SQL/DS

원격 연결 11

CICS/VSE 필요조건 31

CONNECT 명령 12

START 명령

구문 77, 78

오류 디버깅 146

인터페이스 통신 영역 23

키워드

목록 78

DSQADPAN 80

DSQALANG 80

DSQSBSTG 81

DSQSCMD 81

DSQSDBCS 81

DSQSDBNM 82

DSQSDBQN 82

DSQSDBQT 82

DSQSDBUG 82

DSQSDCSS 82

DSQSIROW 82

DSQSMODE 83

DSQSPILL 83

DSQSPRID 83

DSQSRSTG 83

DSQSRUN 83

DSQSSPQN 83

DSQSSUBS 83

DSQSUSER 84

QMF 시동 27, 77

STATE 명령 36

T

TSO

어셈블러 프로그램 168

어셈블러 호출 인터페이스 프로그램

168

TSO (계속)

C 프로그램 186

C 호출 인터페이스 프로그램 186

COBOL 프로그램 204

COBOL 호출 인터페이스 프로그램
204

FORTAN 프로그램 220

FORTAN 호출 인터페이스 프로그
램 220

PL/I 프로그램 237

PL/I 호출 인터페이스 프로그램 237

REXX 프로그램 248

REXX 호출 인터페이스 프로그램
248

V

VSE CICS 137



프로그램 번호: 5675-DB2
5697-F42

SA30-1051-00



Spine information:



QMF

QMF 응용프로그램 개발

버전 7